Mesopotamian journal of Cybersecurity
Vol.2024, **pp**. 17–46
DOI: https://doi.org/10.58496/MJCS/2024/003; ISSN: 2958-6542
https://mesopotamian.press/journals/index.php/cybersecurity

Research Article

A New Tiger Beetle Algorithm for Cybersecurity, Medical Image Segmentation and Other Global Problems Optimization

Ahmed Saihood[1,*], Mohamed Adel Al-Shaher[1], Mohammed A. Fadhel[2]

[1] *College of Computer Science and Mathematics, University of Thi-Qar, Thi Qar, Iraq.*

[2] *College of Computer Science and Information Technology, University of Sumer, Thi Qar, Iraq*

ARTICLE INFO

Article History
Received 10 Feb 2024
Accepted 02 Apr 2024
Published 25 Apr 2024

Keywords
Tiger beetle optimization
TBO
Cybersecurity
Malware detection
Image Segmentation
Reservoir well connectivity

ABSTRACT

The tiger beetle is a fierce and cunning predator insect that uses deception to hunt its prey. The tiger beetle traps and hunts them by digging holes along the path of other insects. This study used a tiger beetle hunting strategy to create a tiger beetle optimization (TBO) algorithm. In this algorithm, each solution represents the position of a tiger beetle, with the optimal position being the prey's location. Using this method, the tiger beetles gradually converge to the optimal solution, creating holes around them and searching for them. We evaluate the TBO algorithm's search capability using a series of well-known mathematical test functions. Moreover, among the sophisticated forms of malware are polymorphic viruses, which are adept at changing their behaviour while maintaining the same essential functions. Thus, a machine learning-based malware analysis system utilizing the power of the proposed TBO is introduced in this article. TBO handles the escalating issues of efficiently catching and mitigating cybersecurity threats in an era where conventional methods suffer from the complexity and volume of modern malware. Compared to other optimization methods, the proposed algorithm has shown less error in finding the optimal solution when implemented and evaluated on different functions. The tiger beetle optimization algorithm has proven helpful in various applications, including image clustering and reservoir well placement, where it can identify damaged areas or tissues with greater accuracy. When diagnosing lung cancer, the proposed method has shown a sensitivity, validity, and accuracy of 88.63%, 87.58%, and 89.86%, respectively, when using the EBT, WKNN, ESKNM, and QSVM methods.

1. INTRODUCTION

Many of the current challenges we encounter can be categorized as optimization problems. Out of numerous potential solutions, only a handful are efficient and can be used to effectively optimize the objective function. These optimization problems have countless applications, such as in engineering, image processing, data mining, and computer networks. One method employed to address optimization problems is through the use of metaheuristic algorithms [1]. These algorithms draw inspiration from natural occurrences and the behavior of living organisms, allowing them to tackle intricate optimization problems. The limitations of metaheuristic algorithms include their simplicity and ability to work without objective function information requirements.

There are various techniques, including Darwin's principle-based evolutionary algorithm (DPA), animal group mimic-based swarm intelligence algorithm (AGA) and nest-building mechanism-based optimization algorithm (NMA). DPA includes genetic programming [3] and genetic algorithms [2]; AGA includes the grey wolf optimizer (GWO) [4], spotted hyena optimizer (SHO) [6], Harris hawks optimizer (HHO) [7], water strider algorithm (WSA) [9], black water window optimization algorithm (BWO) [8] and whale optimization algorithm [5]. The travelling salesperson problem [12] can be solved using the Ant Optimization Algorithm. They present the translation of biological and natural processes into computational procedures that guide the complex landscapes of high-dimensional and nonlinear optimization problems. The HHO algorithm, motivated by Harris' hawks' collective hunting process and outstanding talent, is a dynamic and adaptable method for both exploration and exploitation steps in problem solving. This adaptability is highly influential in determining global optima across diverse problem spaces. Additionally, the BWO algorithm derives its efficiency from simulating the weird mating traditions and survival tactics of black widow spiders, presenting novel crossover and mutation agents that improve its convergence speed and solution accuracy. However, they need help with the possibility of early convergence, mainly in complicated landscapes where the algorithms may become stuck to local optima rather than the global optimum.

*Corresponding author. Email: ahmed.alisiehood@utq.edu.iq

The balance between exploration and exploitation is minimal; excessive exploration can lead to inefficiencies, while excessive exploitation might become stuck in suboptimal solutions.

In computer security, malicious activities carried out by harmful software, scripts, or binary codes aimed at undermining the confidentiality, integrity, and accessibility of data are commonly referred to as malware. When this software compromises private information or data without the owner's knowledge or consent, it breaches confidentiality. The impact on availability occurs when malware disrupts information flow, leading to end users being unable to access data due to server outages or network infrastructure issues. Furthermore, modifying information threatens integrity, as malware can potentially alter data by introducing hazardous code during periods of vulnerability [47].

In cybersecurity, metaheuristics are essential for developing adaptive mechanisms for threat detection and mitigation and delivering robust solutions against sophisticated cyber threats. The healthcare industry uses meta-heuristics in medical image analysis, improving the accuracy of diagnoses, and in pharmaceuticals, where they revolve the drug finding process by swiftly determining potential combinations. Environmental control also involves integrating metaheuristic algorithms, mainly in resource sharing and conservation efforts, such as optimizing water use in agriculture or enhancing the efficiency of renewable energy bases. Furthermore, they aid in network design and optimization in telecommunications, ensuring dedicated and efficient communication infrastructures.

A nonbiological strategy can be utilized to model some algorithms. Gravity or water waves are examples of these phenomena [13]. Furthermore, the human demeanour-based method is another example of such a phenomenon [14],[15],[16],[17].

The existing methods used to enhance malware detection face the challenges posed by large datasets, which often include redundant or irrelevant features that can increase both the misclassification rate and computational time. To address these challenges, malware developers are continually evolving and enhancing their techniques, creating new methods and improving existing methods to circumvent security measures. Recognizing that malware poses one of the most significant security risks to computer systems, it is imperative to swiftly identify and remove such threats to ensure the safety of these systems [48]. The proposed TBO algorithm reduces misclassification rates and computational time, highlighting its feature selection efficiency and potential to facilitate detection.

Optimizing algorithms can be enhanced in several ways. Some algorithms, such as the Harris hawk optimization algorithm, may need to be simplified to be more practical. On the other hand, algorithms such as the black widow optimization algorithm may have superficial connections and not effectively search the problem space. The atom search optimization algorithm may have a high modelling level, but it cannot relate between solutions and does not have an idea about the problem.

Additionally, classifying populations as worthy and unworthy groups may not be wise, as it can result in poor and rich optimization algorithms with incorrect relationships. Algorithms such as the WOA and SHO may look for the optimal solution but can become entangled in local optima. A large population may be needed to prevent the AOA from becoming entangled in optimal paths. The algorithms are restricted to specific search methods. Thus, new methods are required to solve more problems and decrease the shortcomings of known techniques.

Swarm intelligence (SI) optimization algorithms simulate the behaviour of swarms in nature to discover reasonable solutions to complex problems. Each individual in the swarm denotes a candidate solution, and the individuals interact with each other to enhance their solutions over time. Swarm intelligence (SI)-based algorithms have played a key role in many valuable applications, such as the travelling salesperson problem (TSP). For instance, studies have shown that the ant colony optimization (ACO) [18,19] algorithm has a distinctive ability to search for optimal solutions to the TSP.

The no free lunch (NFL) theorem shows that no single optimization algorithm can be universally suitable for all problems. This indicates that while metaheuristic algorithms are very practical and efficient for many optimization problems, there may be some problems where they cannot be used to find the best solution. NFL has motivated the development of more effective optimization algorithms. To address this challenge, we introduce a new SI-based algorithm called the tiger beetle optimizer (TBO). The tiger beetle algorithm is designed to mimic the deceptive and hunting behaviors of these insects, ultimately resulting in optimal solutions that can be applied to a range of scenarios.

To address the multifaceted issues posed by optimization problems across different domains—from cybersecurity to medical image segmentation—our study proposes the tiger beetle optimization (TBO) algorithm. Inspired by the complexity of tiger beetles, the TBO algorithm is a new algorithm for swarm intelligence that includes five distinct updating rules: hunt area selection, hole digging, prey hunting, larvae reproduction, and low-quality hole destruction. These rules collectively facilitate the TBO algorithm to guide the search space efficiently, particularly improving the quality of the solutions discovered.

Our research incidents further theoretical development to practical applications, presenting a machine learning-based system for malware detection (ML-TBO) that capitalizes on the TBO's abilities. This system represents the algorithm's ability to accurately determine and mitigate cybersecurity threats. Additionally, we involve the TBO algorithm in medical image segmentation via a fuzzy C-means clustering process, demonstrating its utility in processing complicated medical data for more reasonable diagnostic outputs. In the domain of petroleum engineering, we examine its application in optimizing

injector–producer connectivity for reservoir well placement, further showing the algorithm's adaptability and efficacy in decoding real-world engineering challenges. The paper's primary contributions include the following:

- The TBO algorithm, a novel swarm intelligence algorithm inspired by the behaviours of tiger beetles, leverages five updating rules to find high-quality solutions: hunt area selection, hole digging, prey hunting, larvae reproduction, and low-quality hole destruction.
- A machine learning-based malware analysis system utilizing the proposed TBO (ML-TBO) is introduced for malware detection as a real-world application for the proposed TBO.
- We present a fuzzy C-means clustering approach based on the proposed TBO algorithm for medical image segmentation.
- Optimizing injector-producer connectivity in Reservoir well placement using the proposed TBO.
- The TBO algorithm is thoroughly evaluated on a suite of mathematical test functions, and the results demonstrate its superior performance compared to that of state-of-the-art optimization techniques.
- The successful application of the TBO algorithm to several practical engineering design problems demonstrates its potential for solving real-world problems.

The rest of this study is organized as follows: Section 2 reviews and models the tiger beetle algorithm's behaviour. Section 3 presents the implementation results. Section 4 investigates the abilities of the proposed method. Section 5 shows how TBO can solve real-world application problems such as malware detection. Finally, the paper provides a conclusion, and conceivable future recommendations are offered in section 6.

2. MATERIALS AND METHODS

It is interesting to see how the tiger beetle optimization algorithm can be used to solve optimization problems. By studying the behavior of this insect, we have been able to formulate an optimal algorithm. This algorithm can then be applied to various optimization problems to find the best solution.

2.1 Inspiration

Understanding the different attributes of different insect species is important. Some insects are known for standing hard-working, while others are described as bloodsuckers. The American beetle, for instance, is known for its strength. Nevertheless, TB is widely regarded as the wildest species due to its aggressive and murderous nature. It is known for its ferociousness towards its target, no matter which group of insects it belongs to. This beetle species is also one of the quickest in the wildlife and is capable of running up to 9 km/h. It can often be seen exploring a hole along its prey's direction and striking it as soon as it spots it. Sometimes, the tiger beetle goes directly for its prey and destroys it with its powerful talons. Another notable attribute of the tiger beetle is its massive and powerful jaws.

Figure 1 presents a glance at the tiger beetle and its hunting features, which make it a fantastic hunter. The hunting mechanism of this species is unclear and intriguing to survey.

The tiger beetle (TB) has a special hunting mechanism, as shown in Figure 1 and Figure 2. They drill holes along the way of their prey and force different insects into these holes to hunt them. Interestingly, any TB can replicate, and the larvae extend in the exact tunnel grid and holes. Larvae are just as hazardous as grown-up TBs. They look around the holes, punching their head out to watch their surroundings with a 360-degree view. As the prey arrives, they open the door to the bottom of the hole. This permits the hunter to carry the prey into the tunnel and successfully hunt it.

Figure 1: A Tiger beetles.

Figure 2: Hiding the tiger beetle in the hole for hunting.

2.2 Modelling the TBO

Several assumptions are outlined below for modelling the TBO algorithm.

- The algorithm characterizes each solution by encoding it a tiger beetle.
- Every hole within the algorithm approximates a distinct region in the objective function, and its fitness is chosen by considering the objective function.
- TB endeavour efforts to exhume holes in regions with a greater possibility of successful hunting.
- Upon finding a hole, a TB estimates it. After that, considering its fitness, the beetle either examines the environment or makes a new hole.
- Every tiger beetle replicates within its specified hole, depositing larvae in its hole and neighboring holes.
- Extra holes are dug close to existing ones, concentrating on areas with an increased probability of good hunting. This strategy acts as a state of local search near the current solution.

1) Producing the initial population

The presented method contains several solutions described by TB. This population is rendered randomly in the problem space, as shown in Equation (1):

$$TB_i^j = L + (U - L).rand \tag{1}$$

The inscription TB_i^j defines the j form of the solution i, approximating the TB. L represents the descending limit further by j, U is the upper boundary in this dimension, and "rand" is a random number within the range (0, 1). The initial population of the TB in the first iteration can be produced using Equation (2):

$$\mathrm{P} = \begin{bmatrix} TB_1^1 & TB_1^2 & \dots & TB_i^D \\ TB_2^1 & TB_2^2 & \dots & TB_2^D \\ \vdots & \vdots & \vdots & \vdots \\ TB_n^1 & TB_n^2 & \dots & TB_n^D \end{bmatrix} \tag{2}$$

In this context, where n denotes the number of TBs and D is the dimension of each solution, each TB is evaluated based on the objective function, represented as F. Equation (3) describes the composition of an array for population evaluation.

$$\mathrm{f(P)} = \begin{bmatrix} f(TB_1^1 & TB_1^2 & \dots & TB_i^D) \\ f(TB_2^1 & TB_2^2 & \dots & TB_2^D) \\ \vdots & \vdots & \vdots & \vdots \\ f(TB_n^1 & TB_n^2 & \dots & TB_n^D) \end{bmatrix} \tag{3}$$

2) Digging hole fitness

The method outlined implicates extemporizing individualized objectives for each TB, seeking to create shelter holes at sites with a greater possibility of successful hunting. One strategy involves increasing the number of holes in near-optimal solutions. An offered tactic entangles choosing the hole number unearthed around the best area using the exponential function e. This is achieved by introducing Equation (4), considering that the goal is to minimize the identified problem. Digging is identical to creating possible solutions (holes) where the algorithm can determine optimal or near-optimal solutions. The "fitness" of these holes defines the grade of the solutions, navigating the algorithm towards more bright areas of the search space. This procedure improves exploration by enabling the algorithm to explore multiple areas and exploit by deepening the search in areas with high potential.

$$H(TB_i) = \left[1 - ex\,p\left(\frac{f(TB_i)}{f(W(t))}\right)\right].H_m \tag{4}$$

$H(TB_i)$ denotes the number of holes bordering TB_i. H_m represents the maximum number of holes that the TB digs into a site. The term $f(TB_i)$ represents the value or fitness of a TB, such as TB_i The parameter H ranges from its maximum value associated with the best TB to a minimum value of zero for the worst TB. This indicates that most holes are dug near the optimal solution to investigate this region also.

3) Dig the hole and positioning in them

In the proposed procedure, each TB_i can tunnel $H(TB_i)$ holes in its surroundings, driven by Equation (4) and contingent on evaluating its fitness. Each beetle can generate offspring in the state of new solutions by making holes, depositing larvae within each hole, and awaiting possible prey. The hole function in the proposed method is outlined as follows:

- Initially, the hole is spread from the central hole with more significant SD, enabling a global search type and stopping the algorithm from becoming entangled in fewer local optima.
- Thereafter, the method transitions slowly to a local search strategy, seeking to underrate computational errors associated with optimization.

A distance function is presented to mitigate the divergence in the standard deviation of the solution's allocation across the Tiger Beetle optimization iterations. Equation (5) suggests the utilization of an arctangent function for this objective:

$$SD(t) = SD_0 - \left|\frac{p}{\pi}\text{atan}(\frac{p}{\pi}t)\right| \tag{5}$$

where p is the scaffold for a coefficient controlling hole reduction and distribution, SD_0 represents the initial standard deviation assigned to 1, and t is the algorithm's iteration counter. Figure *3* shows the SD of the TB hole distribution for three values of p. Raising p broadens the divergence capacity of the SD to [0.8, 1]. When p is equal to 1, the SD ranges between [0.5, 0.9], and for p equal to 0.5, the SD fluctuates within [0.1, 0.65]. To excavate a hole around a given location, Equation (6) can be employed:

$$TB^{new} = \begin{cases} TB^{old} + R.SD(t).rand & r \leq 0.5 \\ TB^{old} - R.SD(t).rand & r > 0.5 \end{cases} \tag{6}$$

In this equation, R denotes the field for breeding the solution and is an arbitrary number within the range [1, 2]. Furthermore, rand is a random number within the interval [0, 1]. The variable r is another random number within [0, 1], specifying whether the solution is bred on a dimension's left or right side.

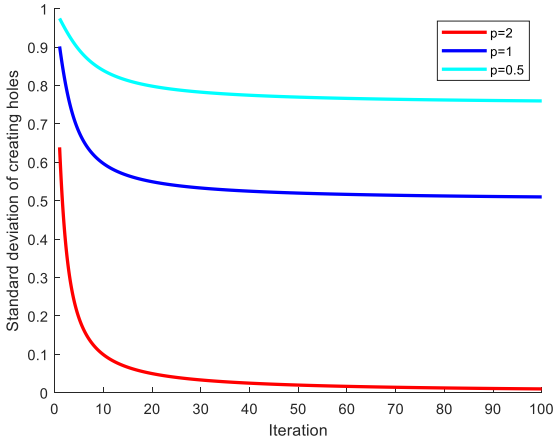

Figure 3: Distribution of hole reduction for chasing prey according to the number of iterations of the tiger beetle algorithm.

4) Hunting and elimination of nonqualified solutions

In the proposed method, holes are tunnelled, and larvae are placed in them. Each hole can catch prey based on its quality, and those in irregular regions may yield during hunting. If a tiger beetle cannot catch any prey in a certain hole, that hole is left or crushed. Some solutions in each iteration are destroyed due to unsuccessful hunting. For this purpose, a mechanism for the chance of survival of a solution is represented through Equation (7) in the context of a minimization problem:

$$p(TB_i) = 1 - \frac{1}{\sum_{i=1}^{n} \frac{f(TB_i)}{f(W(t))}} \times \frac{f(TB_i)}{f(W(t))} \tag{7}$$

In this equation, $p(TB_i)$ denotes the probability of a solution. A random number within the intermission [0, 1], indicated as r, is used to determine whether to destroy or maintain a TB or hole, following Equation (8). P represents the solutions offered by the population used in Equation (8).

$$P = \begin{cases} \{P\} + TB_i & r \le p(TB_i) \\ \{P\} - TB_i & r > p(TB_i) \end{cases} \tag{8}$$

In this scenario, if $r \le p(TB_i)$, the hole and the beetle within it would stay in their existing appointment. On the other hand, if $r > p(TB_i)$, the established beetle will not be considered.

5) Looking in the insect-prone regions

In the suggested procedure, TBs seek to investigate regions with improved chances of hunting insects and developing more holes. A well-designed search function examines the area, including the regions within the population. Equation (9) shows the search mechanism of the proposed algorithm within the region delivering the highest probability of obtaining the optimal solution:

$$TB_i^j = \alpha . TB_i^j + \beta . (B(t) - \bar{B}) . rand \tag{9}$$

In the context of the algorithm, B(t) denotes the role of the best beetle, $\bar{B}$ represents the solution average, α is the reduction coefficient for the current location, and β is the addition coefficient for the existing area. The weights for α and β can be selected using Equations (10) and (11). Figure *4* shows the divergence of these two parameters, showing a reduction and increase over time.

$$\alpha = 1 - (\frac{t}{MaxT})^2 \tag{10}$$

$$\beta = 1 - \alpha \tag{11}$$

where MaxT is the maximum number of iterations of the tiger beetle algorithm and t is the number of iterations of the algorithm.

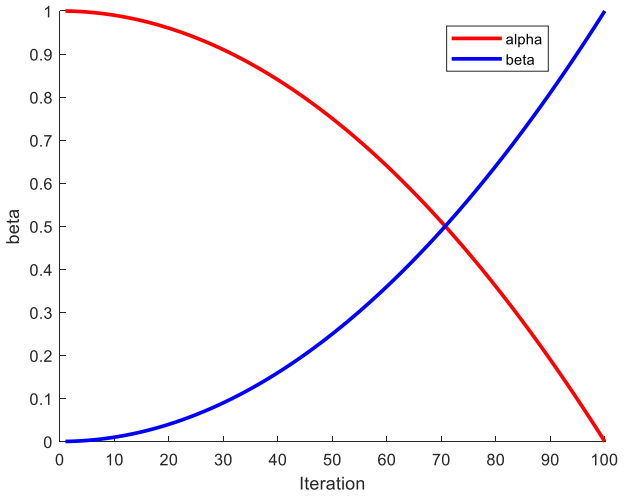

Figure 4: Increase and decrease in α andβ

Towards conclusive iterations, the alpha parameter decreases, reducing the influence of the existing solution. Simultaneously, the beta parameter expands, expanding the investigation range between the mean and the weight.

6) Mating

In the proposed approach, tiger beetles can investigate the problem space in search of appropriate mates. The idea is that TB_i denotes a tiger beetle and can transfer towards the positions of TB_l and TB_k, which are randomly chosen from the population. Mating presents diversity in the population of solutions, assisting in exploration by yielding new solution candidates. Equation (12) expresses the direction of the TB_i beetle in the direction of the two other TBs:

$$TB_i^j = TB_i^j + \left(TB_k^j - TB_l^j\right).rand \tag{12}$$

The choice of dual TBs to be tracked by TB_i the TB affects the behaviour on the one hand and does not become entangled in the local optimum on the other hand.

7) Proposed method algorithm

The pseudocode for the TBO algorithm is given in Algorithm (1). In this proposed strategy, the performance parameters are configured and initialized. After that, the objective function is determined, and a population of solutions, defined as TB, is generated and arranged within the holes in the objective function space. Each TB is evaluated based on the objective function, and the best and worst holes or beetles are identified. Relying on the fitness of each tiger beetle, it can mine multiple holes in its existing place. These holes can be used to yield new larvae. A probability is given to each hole, picking its likelihood of survival or the survival of the beetle within it. At each iteration, less adept solutions may be destroyed.

Pseudocode of Tiger Beetle Optimization(TBO) Algorithm (1)

Input: Objective function & Pop & Parameters
Adjust parameters such as population size(N), number of iterations(MaxT) and number of holes**(**Hmax**)**
for i=1: **N** % Initialize the random population:
  for j=1: **Dim** $\boldsymbol{TB_i^j = L + (U - L).rand}$
  end for
end for
Fill Pop or P Mattix: $\mathbf{P} = \begin{bmatrix} \boldsymbol{TB_1^1} & \boldsymbol{TB_1^2} & \dots & \boldsymbol{TB_i^D} \\ \boldsymbol{TB_2^1} & \boldsymbol{TB_2^2} & \dots & \boldsymbol{TB_2^D} \\ \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{TB_n^1} & \boldsymbol{TB_n^2} & \dots & \boldsymbol{TB_n^D} \end{bmatrix}$
Calculate the fitness values of the population by objective function & t=1, SD0=1; R=2*rand
while(It<=MaxT)
  % Calculation of standard deviation of cavities and alpha and beta coefficients in each iteration
$\boldsymbol{SD(t) = SD_0 - \left|\frac{p}{\pi}\mathbf{atan}(\frac{p}{\pi}t)\right|}$

$\alpha = 1 - (\frac{t}{MaxT})^2$ &$\beta = 1 - \alpha$
%Calculate the Best and worst tiger beetle(B(t) & W(t)) & Calculate the ability to create holes
for i=1: **N**
$H(TB_i) = \left[1 - exp\left(\frac{f(TB_i)}{f(W(t))}\right)\right].H_m$
end for
for i=1: **N**%, Produce several holes and place the larvae inside them as a new solution
for j=1: $H(TB_i)$
if rand>0.5
$TB^{new} = TB^{old} + R.SD(t).rand$
else
$TB^{new} = TB^{old} - R.SD(t).rand$
endif
end for
end for
Sum=0
for i=1: **N**%, Calculate the average position of the tiger beetle
$Sum = Sum + TB_i$
end for
$\overline{B} = \frac{Sum}{N}$
for i=1: **N**%Search for high-insect areas
for j=1: **Dim** $TB_i^j = \alpha.TB_i^j + \beta.(B(t) - \overline{B}).rand$
end for
end for
for i=1: **N** %Mating
Select random two tiger beetle such TB_k &TB_l
for j=1: **Dim** $TB_i^j = TB_i^j + (TB_k^j - TB_l^j).rand$
end for
end for
It=It+1
end while
Output ***B*** or *BestSol*

Considering the proposed method, better solutions have a lower probability of being destroyed. Every conceivable solution has the chance to explore the region between the existing optimum and the centre of the solution's gravity, as this area is more likely to have optimal solutions. A beetle can pick a confidante beetle from the population and follow it for mating. Improving the significance of the proposed approach concerns considering that a beetle desires another beetle only if the beetle promises more experienced mating. In this approach, each beetle randomly chooses two beetles and transits in their directions to improve its exploration ability.

3. IMPLEMENTATION RESULTS

The TBO and other algorithms are implemented on a Windows 10 PC with an Intel Core i7 processor and 32 GB of memory using MATLAB software.

3.1 Evaluation criteria

The TBO algorithm uses multimodal, unimodal, and hybrid evaluation functions.Table 2 and Table 3 delineate the standards for the measurement functions based on their domains and optimal worth. To calculate the performance of the tiger beetle optimization algorithm, 37 well-known algorithms were utilized (*Table 4* to *Table 8*). These functions were organized into unimodal, multimodal, low-dimensional, hybrid, and mixed categories. Unimodal functions (f1-f7) emphasize global optimization without local optima, multimodal functions (f8-f13) feature considerable local optima, low-dimensional functions (f14-f23) include a fixed number of local optima, and hybrid functions (f24-f29) present more random variables. Composite functions (f30-f37) are constructed using subsets of multimodal functions. Hybrid and hybrid-composite functions are regarded as more problematic. The effectiveness of metaheuristic algorithms, including the tiger beetle optimization algorithm, is evaluated based on their error rates in reaching optimal solutions, with lower error rates and decreased trapping in local optima showing higher efficiency in translating optimization problems. Each evaluation function is employed as an objective function in the implementation.

3.2 Implementation parameters

Table *1*demonstrates the implementation parameters and the evaluation of the proposed algorithm alongside similar algorithms.

Table 1: THE PARAMETERS OF THE PROPOSED METHOD AND OTHER METHODS

Algorithm	Parameter	Value	Parameter`	Value
DE	β_{min}	0	β_{max}	2
PSO	c1,c2	2,2	Inertia weight	0.8
BA	A	0.5	f_{max}, f_{min}	2,0
BOA	c	0.1	a	0.01
GWO	c a	0.01 [2,0]	α	0.1
WHO	E	[-2,+2]	r1,r2,r3	[0,1]
FA	α	0.50	β	0.20
BWO	pp	0.6	pm CR	0.4 0.44
ASO	Depth weight	50	Multiplier weight	0.2
TBO	Hm SD0	5-7 0.5-0.9	p	1

Table 2: UNIMODAL, MULTIMODAL, AND LOW-DIMENSIONAL TEST FUNCTIONS

Name	Function	n	Range	Optimum
Sphere	$f_1(x) = \sum_{i=1}^{n} x_i^2$	30	$[-100,100]^n$	0
Schwefel 2.22	$f_2(X) = \sum_{i=1}^{n} \lvert X_i \rvert + \prod_{i=1}^{n} \lvert X_i \rvert$	30	$[-10,10]^n$	0
Schwefel 1.2	$f_3(X) = \sum_{i=1}^{n} \left(\sum_{j-1}^{i} X_j \right)^2$	30	$[-100,100]^n$	0
Schwefel 2.21	$f_4(X) = max\{\lvert X_i \rvert, 1 \le i \le n\}$	30	$[-100,100]^n$	0
Rosenbrock	$f_5(X) = \sum_{i=1}^{n-1} [100(X_{i+1} - X_i^2)^2 + (X_i - 1)^2]$	30	$[-30,30]^n$	0
Step	$f_6(X) = \sum_{i=1}^{n} ([X_i + 0.5])^2$	30	$[-100,100]^n$	0
Quartic	$f_7(X) = \sum_{i=1}^{n} iX_i^4 + random[0,1)$	30	$[-1.28,1.28]^n$	0
Schwefel	$f_8(x) = -\sum_{i=1}^{n} \left(x_i sin\left(\sqrt{\lvert x_i \rvert}\right) \right)$	30	$[-500,500]^n$	−12569.5
Rastrigin	$f_9(x) = \sum_{i=1}^{n} (x_i^2 - 10cos(2\pi x_i) + 10)^2$	30	$[-5.12,5.12]^n$	0
Ackley	$f_{10}(x) = -20exp\left(-0.2 \sqrt{\frac{1}{n}\sum_{i=1}^{n} x_i^2} \right) - exp\left(\frac{1}{n}\sum_{i=1}^{n} cos(2\pi x_i) \right) + 20 + e$	30	$[-32,32]^n$	0
Griewank	$f_{11}(x) = \frac{1}{4000}\sum_{i=1}^{n} (x_i - 100)^2 - \prod_{i=1}^{n} cos\left(\frac{x_{i-100}}{\sqrt{i}}\right) + 1$	30	$[-600,600]^n$	0
Penalized	$f_{12} = \frac{\pi}{n}\left\{10sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2[1 + 10sin^2(\pi y_i + 1)] + (y_n - 1)^2\right\} + \sum_{i=1}^{30} u(x_i, 10,100,4)$	30	$[-50,50]^n$	0
Penalized2	$f_{13}(x) = 0.1\left\{sin^2(3\pi x_1) + \sum_{i=1}^{29} (x_i - 1)^2 p[1 + sin^2(3\pi x_{i+1})] + (x_n - 1)^2[1 + sin^2(2\pi x_{30})]\right\} + \sum_{i=1}^{30} u(x_i, 5,10,4)$	30	$[-50,50]^n$	0
Foxhole	$f_{14}(x) = \left(\frac{1}{500} + \sum_{j=1}^{25} \frac{1}{j + \sum_{i=1}^{2} (x_i - a_{ij})^6} \right)^{-1}$	2	$[-65.536, 65.536]^n$	0.998

Kowalik	$f_{15}(x) = \sum_{i=1}^{11}\left[a_i - \frac{x_1(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4}\right]^2$	4	$[-5, 5]^n$	3.075×10^{-4}
Six Hump Camel	$f_{16}(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	2	$[-5, 5]^n$	-1.0316
Branin	$f_{17}(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)cosx_1 + 10$	2	$[-5, 10] \times [0, 15]$	0.398
Goldstein-Price	$f_{18}(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 + 1 - 3x_2)^2 (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	2	$[-2, 2]^n$	3
Hartman 3	$f_{19}(x) = -\sum_{i=1}^{4} exp\left[-\sum_{j=1}^{3} a_{ij}(x_j - p_{ij})^2\right]$	3	$[0, 1]^n$	-3.86
Hartman 6	$f_{20}(x) = -\sum_{i=1}^{4} exp\left[-\sum_{j=1}^{6} a_{ij}(x_j - p_{ij})^2\right]$	6	$[0, 1]^n$	-3.322
Shekel 5	$f_{21}(x) = -\sum_{i=1}^{5} \lvert(X - a_i)(X - a_i)^T + c_i\rvert^{-1}$	4	$[0, 10]^n$	-10.1532
Shekel 7	$f_{22}(x) = -\sum_{i=1}^{7} \lvert(X - a_i)(X - a_i)^T + c_i\rvert^{-1}$	4	$[0, 10]^n$	-10.4028
Shekel 10	$f_{23}(x) = -\sum_{i=1}^{10} \lvert(X - a_i)(X - a_i)^T + c_i\rvert^{-1}$	4	$[0, 10]^n$	-10.5363

Table 3: HYBRID AND COMPOSITION TEST FUNCTIONS

Name	**Function**	***n***	**Range**	**Optimum**
$f_{24}(x)$	Hybrid Function 1($N = 3$)	30	$[-100,100]^n$	1700
$f_{25}(x)$	Hybrid Function 2 ($N = 3$)	30	$[-100,100]^n$	1800
$f_{26}(x)$	Hybrid Function 3 ($N = 4$)	30	$[-100,100]^n$	1900
$f_{27}(x)$	Hybrid Function 4 ($N = 4$)	30	$[-100,100]^n$	2000
$f_{28}(x)$	Hybrid Function 5($N = 5$)	30	$[-100,100]^n$	2100
$f_{29}(x)$	Hybrid Function 6 ($N = 5$)	30	$[-100,100]^n$	2200
$f_{30}(x)$	Composition Function 1 ($N = 5$)	30	$[-100,100]^n$	2300
$f_{31}(x)$	Composition Function 2 ($N = 3$)	30	$[-100,100]^n$	2400
$f_{32}(x)$	Composition Function 3 ($N = 3$)	30	$[-100,100]^n$	2500
$f_{33}(x)$	Composition Function 4 ($N = 5$)	30	$[-100,100]^n$	2600
$f_{34}(x)$	Composition Function 5 ($N = 5$)	30	$[-100,100]^n$	2700
$f_{35}(x)$	Composition Function 6 ($N = 5$)	30	$[-100,100]^n$	2800
$f_{36}(x)$	Composition Function 7 ($N = 3$)	30	$[-100,100]^n$	2900
$f_{37}(x)$	Composition Function 8 ($N = 3$)	30	$[-100,100]^n$	3000

Several algorithms have been utilized for comparison, including the differential evolution algorithm and particle swarm optimization (PSO) algorithm. We meticulously describe the parameters and operational stages of the TBO algorithm alongside different methods. For each technique, vital parameters such as DE's $\beta_{min}$and β_{max}, PSO's cognitive and social coefficients (c1,c2) along with its inertia weight, and the specific characteristics of GWO, including its adaptive coefficients (c,a), are distinctly explained. For TBO, parameters such as hunt maturity (Hm), the initial search domain (SD0), and the probability of exploration (p) are emphasized to describe the algorithm's fine method for balancing exploration and exploitation. These parameters, established on comprehensive experimentation, are essential to the operation of each algorithm.

4. INVESTIGATING TBO CAPABILITY

F1-F7 are unimodal test functions, meaning that they have only one global best solution. They are widely used in the swarm intelligence optimization community to evaluate the exploitation capability of algorithms. Table *4* shows the average, rank and standard deviation of the fitness values for the TBO algorithm on these functions. Table *4* shows that the TBO algorithm outperforms the other seven optimization algorithms on the classical test functions F1-F3, F5 and F7, as measured by the average, rank and standard deviation. Specifically, the TBO algorithm's mean fitness value is closer to the theoretical optimum than the other algorithms for the mentioned functions, indicating that the TBO algorithm has high exploitation ability. While the TBO algorithm does not achieve the best average fitness value for F4 and F6, it still ranks second behind the DBO algorithm.

Table 4: COMPARISONS OF RESULTS FOR UNIMODAL FUNCTIONS

F	Metric	TBO	BAT	PSO	DE	FA	BOA	GWO	WHO	BWO	ASO
F1	AVG	**1.46E−14**	1.06E−4	5.23E−4	1.0069	5.82564	1.22E−4	7.33E−8	2.69E−7	1.55E−8	3.45E−12
	STD	**4.56E−7**	2.88E−4	7.55E−4	0.8753	2.99856	8.23E−3	1.72E−4	2.95E−6	1.12E−6	8.28E−6
	RANK	**1.27**	6.85	6.92	7.69	9.52	5.86	4.53	4.87	3.17	2.14
F2	AVG	**2.86E−12**	9.5682	2.83E−4	1.8536	5.7523	0.02569	3.91E−9	5.29E−11	6.33E−10	5.12E−10
	STD	**1.06E−9**	5.1568	3.72E−4	0.7853	0.5692	4.11E−3	6.03E−5	4.39E−8	6.54E−6	5.07E−8
	RANK	**1.12**	9.34	5.86	7.82	8.16	6.72	4.53	2.18	3.78	2.74
F3	AVG	**0.007625**	15.6952	185.1475	28.8536	36.0263	0.00898	0.09025	0.09920	9.68531	83.6954
	STD	0.563227	5.00258	120.4269	5.75863	12.6359	0.086473	0.008632	**0.001563**	0.96635	52.6986
	RANK	**2.19**	6.97	8.23	7.53	8.07	2.84	3.77	4.18	6.42	7.76
F4	AVG	3.83E−09	1.003695	12.14402	1.230064	1.00691	0.008631	2.39E−06	**6.56E−10**	7.76E−09	2.65E−08
	STD	6.73E−05	0.058625	10.75263	0.52392	0.22653	7.49E−03	8.23E−03	6.55E−07	**2.56E−07**	9.16E−06
	RANK	**1.16**	5.89	7.53	6.75	5.93	4.55	3.67	1.28	1.83	2.65
F5	AVG	**21.823**	121.6963	113.8531	189.3269	82.0983	160.61	26.0334	24.14362	28.8536	22.5863
	STD	**0.40369**	132.4621	136.8314	144.112	156.057	120.8543	2.22361	0.4568	0.68532	0.86953
	RANK	**2.26**	6.23	5.79	7.06	5.36	6.84	3.88	2.92	3.76	2.73
F6	AVG	**1.08E−19**	12.02631	0.09522	21.0961	26.48632	1.50265	6.59E−16	1.66E−18	2.56E−18	2.56E−19
	STD	9.23E−11	0.88237	0.002387	1.44603	5.63409	0.02365	1.77E−11	**1.24E−12**	5.88E−12	6.74E−11
	RANK	2.34	6.16	4.88	7.18	7.42	5.56	3.82	2.77	2.54	**2.18**
F7	AVG	**0.01153**	0.039856	0.077362	0.079362	0.01169	0.076329	0.03056	0.02963	0.02632	0.039526
	STD	**0.00914**	0.086362	0.016026	0.013652	0.86392	0.028623	0.086234	0.04436	0.01526	0.014269
	RANK	**2.62**	4.66	5.34	5.68	2.97	5.19	4.16	3.25	3.23	4.29

Table 5: COMPARISONS OF RESULTS FOR MULTIMODAL FUNCTIONS

F	Metric	TBO	BAT	PSO	DE	FA	BOA	GWO	WHO	BWO	ASO
F8	AVG	**-9273.23**	−5680.93	−5207.44	−5032.98	−5414.83	−5583.14	−5672.37	−7521.66	−7521.66	**−7432.18**
	STD	286.265	632.11	506.16	703.18	563.53	486.07	533.98	388.16	488.98	**157.1483**
	RANK	**1.63**	3.67	6.12	6.41	5.86	5.69	3.88	2.24	2.39	**1.88**
F9	AVG	5.03E−16	4.03E-4	3.37E−9	1.44E-3	3.93E-6	6.17E-12	2.57E−10	1.86E−10	4.12E−16	**1.47E−8**
	STD	**4.08E−6**	8.16E−02	7.91E−3	5.53E−03	1.3621	6.09E−04	5.92E−3	9.11E−4	3.76E−5	**5.66E−4**
	RANK	**1.22**	6.69	4.37	6.82	5.83	3.72	3.96	3.92	1.26	**4.64**
F10	AVG	**4.63E−13**	0.00439	5.79E−7	0.05369	0.00892	6.33E−8	6.81E−9	3.93E−9	9.11E−6	**2.95E−11**
	STD	6.73E−5	0.08615	2.64E−4	0.80261	0.94627	9.06E−5	**6.86E−8**	7.44E−7	8.16E−5	**4.69E−7**
	RANK	**1.98**	5.83	4.65	6.26	6.18	3.83	2.76	3.18	4.84	**2.17**
F11	AVG	**8.94E−16**	3.64E−6	8.94E−10	0.01793	364E−8	0.01683	8.23E−9	8.94E−15	8.54E−5	**9.16E−9**
	STD	**6.18E−10**	8.76E−3	8.63E−5	0.05636	1.29E−5	0.96315	1.92E−6	**3.14E−12**	5.11E−3	**2.08E−6**
	RANK	**1.67**	3.64	2.09	5.66	2.83	6.83	2.48	1.75	4.09	**2.44**
F12	AVG	**3.78E−24**	0.036982	0.224573	1.29836	9.01E−9	6.08E−11	2.78E−11	2.44E−22	6.08E−20	***5.14E−23***
	STD	2.44E−16	0.864231	0.510369	0.36952	5.69E−5	2.59E−6	6.11E−8	**5.02E−18**	2.16E−16	**6.54E−17**
	RANK	1.86	5.66	6.16	6.73	4.84	4.11	3.86	2.43	3.42	**1.74**
F13	AVG	**8.16E−23**	0.039206	0.18693	0.33582	0.07362	6.86E−15	5.88E−16	8.32E−21	**8.92E−23**	**6.16E−22**
	STD	8.41E−12	0.001755	0.89032	0.77361	0.06695	2.08E−11	6.94E−11	**5.16E−20**	9.74E−11	**6.67E−15**
	RANK	1.48	4.63	5.38	6.93	4.81	3.49	3.23	2.93	**1.36**	**2.21**

We prioritized three required parameters—computation error, standard deviation, and algorithm rank—each of which provides a unique understanding of algorithm performance. Computational error, calculated as the conclusive distinction between the algorithm's output and the known optimal solution, is a straightforward measurement of accuracy, emphasizing the algorithm's precision in determining optimal solutions. The standard deviation, which is derived from the variability in results across numerous runs, distinguishes the algorithm's consistency and reliability under various initial states. Finally,

the algorithm rank determined via a comparative analysis of performance metrics across all algorithms under consideration feeds a hierarchical positioning that recalls each algorithm's general efficacy in solving optimization problems. These parameters collectively suggest a multifaceted evaluation framework highlighting quantitative performance and facilitating a qualitative comparison among competing algorithms.

We approximate the performance of different optimization algorithms, including TBO, across standardized unimodal functions. The effectiveness of each algorithm is quantitatively evaluated via parameters such as the average computation error (AVG), standard deviation (STD), and algorithm rank (RANK) for each function (F). The AVG computation error provides insight into the algorithm's accuracy, demonstrating its proximity to the optimal solution. The STD contemplates the consistency of the algorithm's performance, conveying an awareness of its reliability across numerous runs. Finally, the RANK algorithm, resolved via a comparative analysis of these metrics, shows a significant difference, with lower values indicating excellent performance. This systematic evaluation emphasizes the TBO algorithm's outstanding precision and robustness, as evidenced by its highest ranks and smallest computational errors. It also highlights the importance of these metrics in discerning each algorithm's nuanced capabilities within the landscape of optimization challenges.

Table 6: COMPARISONS OF RESULTS FOR LOW-DIMENSIONAL FUNCTIONS

F	Metric	TBO	BAT	PSO	DE	FA	BOA	GWO	WHO	BWO	ASO
F14	AVG	**0.998692**	16.35	1.4832	1.6852	3.8756	2.8957	4.2631	1.08526	1.022847	**0.997632**
	STD	1.63E−10	6.6352	0.4886	0.9862	2.1836	1.6695	3.5769	0.92654	**6.76E−15**	**7.40E−11**
	RANK	**1.08**	6.48	3.27	3.46	4.86	4.55	5.06	2.24	1.98	**1.18**
F15	AVG	**4.11E−06**	3.46E−02	2.99E−04	5.54E−03	7.56E−04	6.19E−03	1.12E−03	2.95E−04	6.17E−04	**5.88E−05**
	STD	**6.17E−05**	3.97E−03	5.46E−03	2.33E−03	1.61E−03	5.77E−03	9.01E−03	1.09E−04	6.99E−04	**4.95E−04**
	RANK	2.73	6.18	3.11	4.83	3.44	4.29	4.18	2.96	3.28	**2.59**
F16	AVG	**−1.03163**	−6.67E−1	−1.03E+0	−1.0E+0	−1.03E+0	−1.03E+0	−1.03E+0	−1.03103	**−1.03163**	**−1.03163**
	STD	**6.14E−16**	0.8962	1.86E−03	1.15E−14	1.03E−15	9.02E−15	8.61E−16	7.93E−15	1.83E−15	**3.55E−15**
	RANK	**1.16**	5.02	4.74	3.73	4.66	3.18	2.14	1.39	1.22	**1.28**
F17	AVG	**0.397844**	3.98E−01	4.00E−01	3.98E−01	3.98E−01	3.98E−01	3.98E−01	3.98E−01	3.98E−01	**0.397892**
	STD	**0**	5.44E−5	7.01E−06	1.14E−8	8.06E−15	4.66E−15	6.77E−15	6.19E−12	9.63E−08	**0**
	RANK	**1**	4.73	4.25	4.18	3.58	3.18	3.42	3.89	4.12	**2**
F18	AVG	**3.00E+00**	1.26E+01	**3.10E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**	**3.03E+00**	**3.00E+00**	**3.00E+00**	**3.00E+00**
	STD	**0**	23.58	2.98E−03	**0**	**0**	**0**	5.66E−08	**0**	**0**	**5.68E−18**
	RANK	**1.06**	6.94	5.63	1.17	1.23	1.19	1.68	1.34	1.76	**1.54**
F19	AVG	**−3.86442**	−3.844	−3.861	−3.861	−3.8641	−3.8626	−3.8608	−3.8619	−3.862	**−3.86341**
	STD	5.23E−11	0.14826	4.52E−04	4.11E−13	**3.16E−15**	8.16E−03	5.98E−04	5.43E−04	0.00586	**3.92E−12**
	RANK	1.28	2.44	2.27	3.19	3.42	3.87	4.66	4.53	4.96	**1.14**
F20	AVG	**−3.322**	−3.2644	−3.1405	−3.2682	−3.27088	−3.24922	**−3.322**	**−3.322**	**−3.322**	**−3.322**
	STD	**2.01E−15**	5.44E−07	9.01E−06	6.98E−05	0.008292	0.028146	6.11E−15	8.07E−14	6.08E−14	**2.17E−14**
	RANK	1.46	2.67	4.81	2.54	2.33	3.52	1.37	1.58	1.87	**1.32**
F21	AVG	−9.61698	−4.4406	−5.8856	−9.63629	−8.53264	−7.82056	−8.78265	**−10.1688**	−10.126	**−8.78362**
	STD	0.23682	1.6692	1.28655	1.088733	2.0895	4.32156	3.85649	**0.0725614**	0.43586	**2.47263**
	RANK	2.91	4.52	3.96	2.19	2.86	3.42	2.72	1.57	1.83	**−10.1532**
F22	AVG	**−10.4029**	−6.00551	−7.53061	−9.88036	−9.78226	−8.90265	−10.4022	−10.4018	**−10.4011**	**−10.4029**
	STD	**2.76E−14**	14.56214	2.66381	0.98025	1.08621	**9.01E−4**	**1.04E−4**	1.03695	**7.37E−8**	**6.52E−13**
	RANK	**2.17**	5.88	4.09	3.24	3.38	−9.152336	2.38	2.44	2.87	**2.34**
F23	AVG	**−10.5365**	−6.1206	−6.63584	−10.5018	−9.88263	−8.58233	−10.0986	**−10.5361**	−10.0344	**−10.5317**
	STD	5.02E−12	4.33156	3.84726	3.07E−10	2.86234	1.869422	0.084623	0.00526	0.927655	**6.17E−14**
	RANK	**1.33**	4.92	4.28	2.86	3.83	4.05	2.71	1.42	2.62	**1.46**

Table 7: COMPARISONS OF RESULTS FOR HYBRID FUNCTIONS

F	Metric	TBO	BAT	PSO	DE	FA	BOA	GWO	WHO	BWO	ASO
F24	AVG	**396.17**	1286.365	751.367	429.558	469.661	411.125	484.592	399.189	397.7826	**406.746**
	STD	43.8625	148.568	72.586	61.872	249.366	66.7068	141.026	82.694	69.23594	**15.6321**
	RANK	**1.28**	7.61	5.38	3.31	3.62	2.83	476.1256	1.87	1.66	**2.43**
F25	AVG	**908.159**	1262.36	1026.44	916.026	951.526	946.9825	985.635	910.08	912.782	**910.305**
	STD	5.11E−05	48.621	21.038	0.98264	12.8423	24.2569	25.6215	9.15E−07	8.03E−06	**3.92E−12**
	RANK	1.63	6.78	5.61	2.67	3.26	4.08	3.84	1.97	2.09	**1.85**
F26	AVG	910.853	1476.023	1105.81	912.26	938.89	924.068	962.218	**910**	910.982	**1911.82**
	STD	0.006821	38.1406	28.153	0.51236	12.004	18.9026	20.4193	**0.00658**	0.025916	**0.62581**
	RANK	**1.27**	6.81	5.09	2.85	4.38	3.92	4.11	1.42	2.41	**2.77**
F27	AVG	**910**	1436.841	1188.231	916.122	946.6695	927.4602	929.8034	**910**	914.521	**910.143**
	STD	2.27E−16	58.5626	16.52138	0.001864	10.036526	0.00623	9.03E−04	6.82E−14	2.77E−09	**8.07E−16**
	RANK	1.24	5.11	4.78	2.58	4.51	2.69	3.14	1.38	2.21	**1.89**
F28	AVG	861.256	1988.5452	1706.369	1501.631	988.5241	1403.811	1411.368	**860.2561**	862.819	**874.0214**
	STD	0.02644	545.326	21.2586	55.317	185.369	14.5217	59.5463	0.862357	0.69531	**0.02401**
	RANK	**1.38**	6.88	6.67	5.84	3.83	4.51	4.63	**1.24**	2.89	**3.05**
F29	AVG	**558.1023**	2202.861	2088.143	1811.441	1952.361	1806.641	1898.561	559.0268	558.7703	**558.8863**
	STD	**0.05746**	25.62147	32.4156	8.9513	6.8418	4.8513	11.2576	12.06842	0.53621	**0.088712**
	RANK	**1.25**	6.74	6.14	4.51	5.12	3.87	4.83	2.09	1.46	**1.74**

Unlike unimodal functions with one global best solution, multimodal functions have many local minima, making them difficult to optimize. As the dimension of the search space increases, the number of local optima in multimodal functions grows exponentially. Therefore, multimodal functions are useful for evaluating the exploration ability of optimization algorithms. Table *4* shows the results of the TBO algorithm on six multimodal test functions. For fixed-dimensional multimodal functions, the performances of all the algorithms are similar, but the TBO algorithm remains competitive.

4.1 Time complexity analysis

The computational complexity of the TBO algorithm is affected by three primary operations: the generation of the initial population, the evaluation of solution fitness, and the updating of the TB' state. Considering that N, T, and D represent the initial population size, maximum number of iterations, and number of dimensions, respectively, the sophistication of population generation is provided by N×D. In the updating phase across straight iterations, the complexity is represented as O(T×(4×N×D+N×D×H_{max})). It is essential to emphasize that coefficient 4 is utilized because the population experiences at least four differences in each iteration. The term O (N× D) can be missed, resulting in the computational complexity of the TBO algorithm being marked as O (T× (4×N×D+N×D×H_{max})).

4.2 Evaluation of error and convergence

A preferred evaluation function can experience judgments pursued by implementing TBO and other metaheuristic algorithms. The resulting average optimal computation error rates across iterations are then shown in the output. Figure 5 (a-f) illustrates the application of the proposed algorithm and other metaheuristic algorithms on representative evaluation functions. These graphs consider uniformity in the population size, number of dimensions, and iteration number, which are set at 3, 10, and 100, respectively. Furthermore, each experiment was iterated 30 times. The graphs show six representative evaluation functions, permitting a comparison of the optimal computation error rates over iterations between TBO and other metaheuristic algorithms. Notably, the results indicate that the average global error rate of the optimal estimation for the tiger beetle algorithm is consistently lower than that for the alternatives. Further analysis indicated a significant reduction in the optimal estimation error rate within the TBO.

A straight reduction in the error rate implies that the TBO efficiently instructs solution populations towards the optimal solution with less iteration and acceleration. Especially significant is the observation that the TBO algorithm's error rate in

the last iteration is less than that of most of the approximated algorithms. In specific combined metaheuristic algorithms, such as Ackley, the deduction slope of the error rate is moderately satisfactory. This tolerable squeeze can be attributed to some experiments entangling the algorithm to local optima. The unique attribute of the TBO, wherein it permanently lowers the error slope across different functions, offers its strength against being trapped in local optima. This highlights its ability to bypass local optima and its high brightness in guiding optimal solutions.

4.3 Statistical tests

For the statistical analysis of the TBO and further approximated algorithms, three parameters are used: the computation error of the global optimum on average, the standard deviation (SD) of tests, and the algorithm's rank in discovering the optimal solution. The results for low-dimensional, unimodal, and multimodal test functions and combined functions are presented in Table *4* and Table *5*. Each investigation strengthens a population size of ten and spans 100 iterations, with the average error rate describing the global optimal estimation error in the last iteration of the algorithms. A lower average error rate indicates more unrealistic accuracy, and the proposed TBO produces a lower error rate for 29 out of 37 evaluation functions. This suggests that the TBO shows a lower error in 78.37% of the cases when corresponding to other algorithms. Regarding the standard deviation, a vital metric for algorithm evaluation, the TBO always has a lower standard deviation. A more minor standard deviation indicates a more stable algorithm for finding the optimal solution. The discussion on reducing the standard deviation in the TBO for evaluating its strength is magnified further in the subsequent sections. The Wilcoxon rank test is used in the investigations to evaluate the algorithms.

In this examination, when the propensity of the chosen number is 1, the algorithm minimizes errors in most experiments. Accordingly, the TBO always ranks first in most evaluation functions. For functions 1-7 (Table *4*), the TBO surpassed the others in 85.71% of the cases. For evaluation functions 8-13 (Table *5*), the proposed algorithm achieved the best ranking in 66.66% of the cases. For functions 14-23 (Table *6*), the proposed method ranks 60% more reasonably than the other methods. Finally, for functions 24 to 29 (Table *7*), the proposed algorithm achieved the highest performance in 66.67\% of the cases compared to the other algorithms.

The TBO algorithm consistently shows lower standard deviations, highlighting its stability and reliability in choosing optimal solutions across repeated tests. The reduction in standard deviation with TBO, which is expressive of its robust performance, is examined in depth in later sections. According to the Wilcoxon rank-sum test for algorithm comparison, the TBO algorithm consistently outperforms the other algorithms in 85.71% of the cases for functions 1-7 and 66.66% for functions 8-13 and maintains strong performance across all the considered functions. This analysis highlights the effectiveness of TBO in different scenarios and its potential as a reliable and accurate tool for overcoming optimization challenges.

The results highlight the exceptional precision and robustness of TBO for hybrid functions. It consistently outperforms other algorithms with the lowest average error rates and standard deviations, highlighting its excellent optimization ability in complex landscapes. TBO performed the best on functions F24 and F25, demonstrating its exceptional performance in guiding multimodal functions with complex global and local optima.
Similarly, TBO exhibited outstanding performance in the composition functions, mainly in functions F30 and F31. Its average scores and tiniest standard deviations emphasize its significance in solving composition challenges, which are known for their complexity due to mixing several benchmark functions into a single optimization problem.

For functions 30-37 (Table *8*), the suggested algorithm achieves better rankings in more than 50% of the cases. Across evaluation functions 1-37, the TBO algorithm always shows higher accuracy than the atom search algorithm. Although Black Widow outperforms the suggested algorithm in terms of obtaining the optimal solution in several examples, a comprehensive examination and investigation of the ranks of these algorithms across dimensions would help evaluate the stability of the proposed algorithm and other algorithms.

In Figure *6*, the average rank of the presented algorithm across three dimensions (50, 100, and 200) is displayed regarding the mean error index compared to other algorithms. Figure *7* further depicts the rank of the proposed algorithm and different metaheuristic algorithms established on the standard deviation index. An all-around investigation of the graphs indicates that the presented algorithm always achieves an outstanding rank in computing errors and standard deviations across 50, 100, and 200 dimensions. The higher rank of the suggested algorithm in the average optimal mistake represents its improved accuracy. Moreover, its superior ranking in standard deviation highlights the embellished stability of TBO when handling problems within multidimensional spaces.

The statistically significant p values acquired from t tests—0.0012 for cybersecurity and data protection, 0.004 for medical image segmentation, and 0.012 for reservoir well placement—show that the TBO algorithm significantly improved the ability to solve real-world optimization problems across these domains. These results characterize the outstanding performance of TBO over traditional algorithms and highlight its applicability and usefulness in specific and critical fields. The low p values suggest that the advancements attributed to the TBO algorithm are highly unlikely due to random chance. This highlights its potential as a universal and effective tool for tackling complex cybersecurity, medical image analysis, and energy resource management optimization challenges.

Table 8: COMPARISONS OF RESULTS FOR COMPOSITION FUNCTIONS

F	Metric	TBO	BAT	PSO	DE	FA	BOA	GWO	WHO	BWO	ASO
F30	AVG	**2536.412**	2841.3	2697.568	2557.526	3412.18	2751.058	2651.156	2539.548	2537.895	2608.651
	STD	0.52361	25.6791	18.8146	0.85962	19.5628	16.5421	1.14236	**0.05132**	1.05864	1.14621
	RANK	**1.64**	5.66	4.37	2.29	5.06	3.65	2.84	1.86	1.77	1.97
F31	AVG	**2598.1405**	2705.684	2644.681	2873.66	2792.34	2658.514	2855.695	2601.667	2598.654	2633.171
	STD	**0.058836**	12.0984	9.8065	12.086	6.67026	0.9806	4.25902	0.09842	0.12074	2.66530
	RANK	1.86	3.86	2.49	5.82	4.26	2.53	6.09	1.68	**1.54**	1.62
F32	AVG	2702.541	2882.142	2726.6905	2725.618	2720.816	2708.853	2711.0162	2704.006	2703.1452	2701.453
	STD	**0.01548**	1.2514	9.0048	1.8362	2.56148	1.5257	2.08433	2.6582	0.99563	1.0981
	RANK	1.59	5.38	4.73	4.68	4.11	2.93	3.68	2.29	1.77	1.48
F33	AVG	**2700.012**	2788.561	2708.517	2716.504	2711.2561	2728.18	2704.493	2701.453	2709.542	2701.089
	STD	1.5843	14.5913	12.5873	3.6614	2.5247	9.0145	1.9832	**0.069523**	3.71695	0.90654
	RANK	**1.89**	5.06	4.25	2.76	2.17	2705.842	2.44	2.06	3.1435	1.72
F34	AVG	3142.1627	3253.4716	3377.946	3297.1536	3236.6715	3416.0257	3332.9432	3136.9826	3134.443	**3124.164**
	STD	12.5423	35.54933	18.2098	33.3541	16.8463	3.67452	8.2657	**1.6527**	10.2546	28.6217
	RANK	2.24	5.61	4.58	3.47	3.18	3.69	2.09	2.14	2.71	2.07
F35	AVG	4001.984	6238.523	7045.578	5028.4261	5544.8432	3872.2145	3888.264	**3845.324**	3866.1452	4795.1736
	STD	121.5463	63.3584	251.3745	207.4653	1596.64125	153.54126	88.64215	98.2545	**15.5432**	286.4972
	RANK	2.44	5.12	6.85	4.69	4.87	2.48	2.51	**1.22**	1.26	3.83
F36	AVG	**7365.253**	12125.802	13774.36	14023.842	10087.364	8984.3694	8859.4826	7446.0267	7451.7163	7367.5803
	STD	**52.4618**	186.945	152.264	102.564	96.6745	124.5486	134.5843	62.3478	83.2564	112.543
	RANK	**2.18**	6.93	6.87	5.52	4.88	4.52	3.83	2.88	3.42	2.57
F37	AVG	**11734.68**	16446.621	148895.36	34563.806	15966.34	22715.36	16364.153	18247.941	1499.433	12012.96
	STD	856.321	1563.412	1705.67	2543.145	**486.341**	2594.36	599.842	1352.49	1605.34	1597.264
	RANK	**2.47**	6.79	6.38	5.98	3.24	4.48	3.77	4.86	3.23	**2.58**

4.4 Iteration-based search analysis

One of the essential approaches for evaluating metaheuristic algorithms, including the tiger beetle algorithm, is the utilization of history diagrams. Figure 8 displays the behaviour of the tiger beetle algorithm in the first and middle iterations for evaluation functions F1, F9, F10, and F11. In the first iteration, population members are dispersed in the random

problem space. In the next iterations, the tiger beetle algorithm approaches the global optimum at the coordinates x = 0 and y = 0. In complex functions such as F9, F10, and F11, the algorithm recognizes local optima and instructs the population toward the global optimum solution.

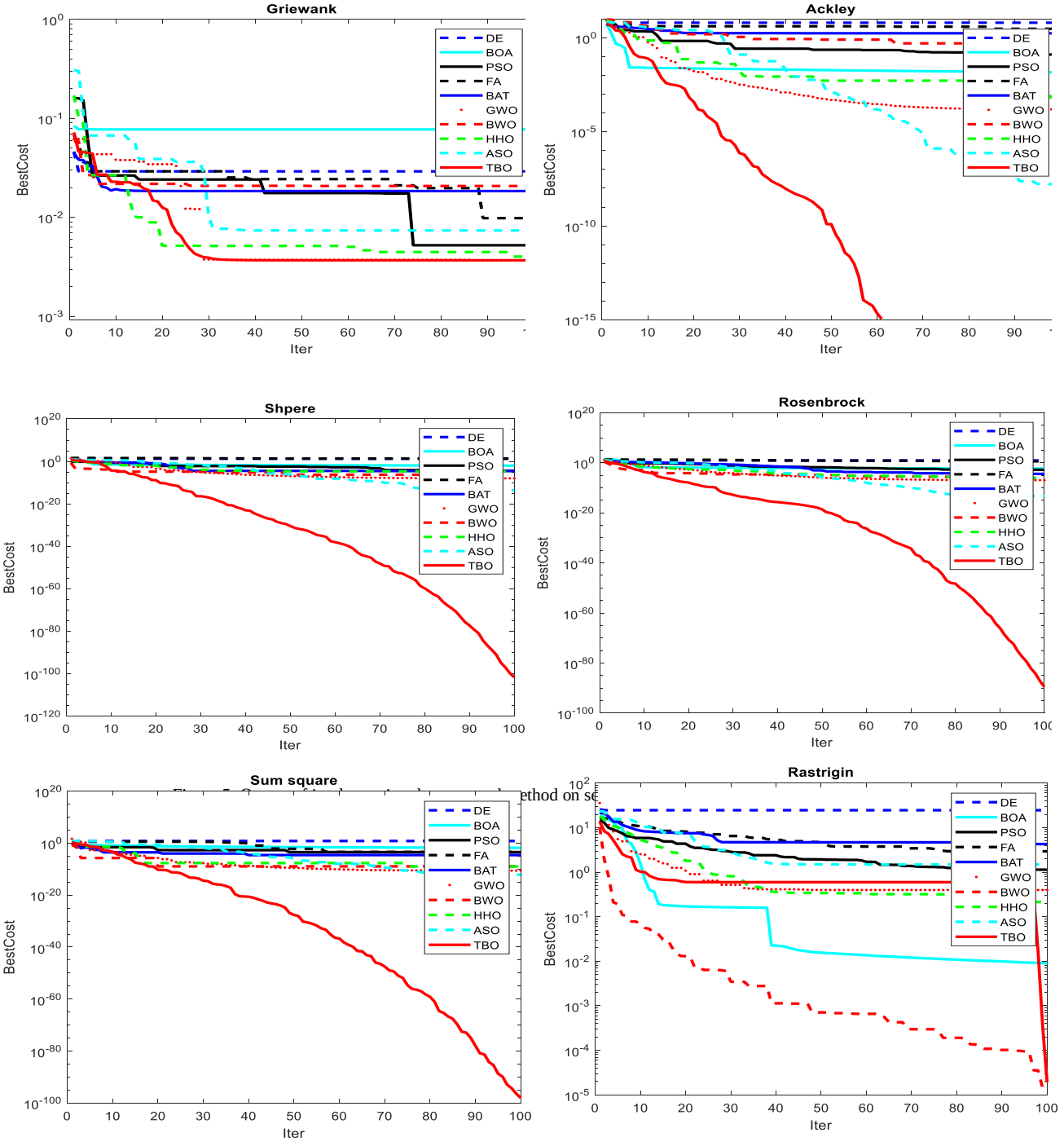

Figure 5: Output of implementing the proposed method on several evaluation functions in 3D mode.

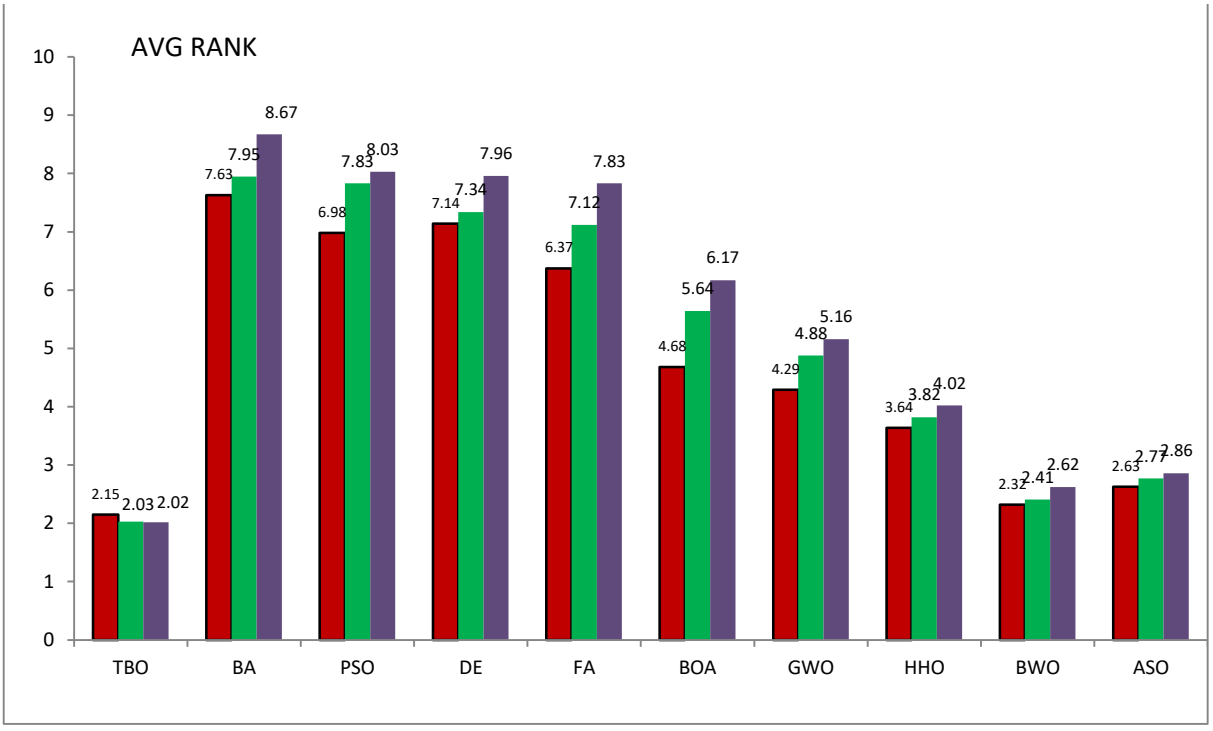

Figure 6: Evaluation of the presented algorithm's ranking in terms of error estimation relative to other algorithms.

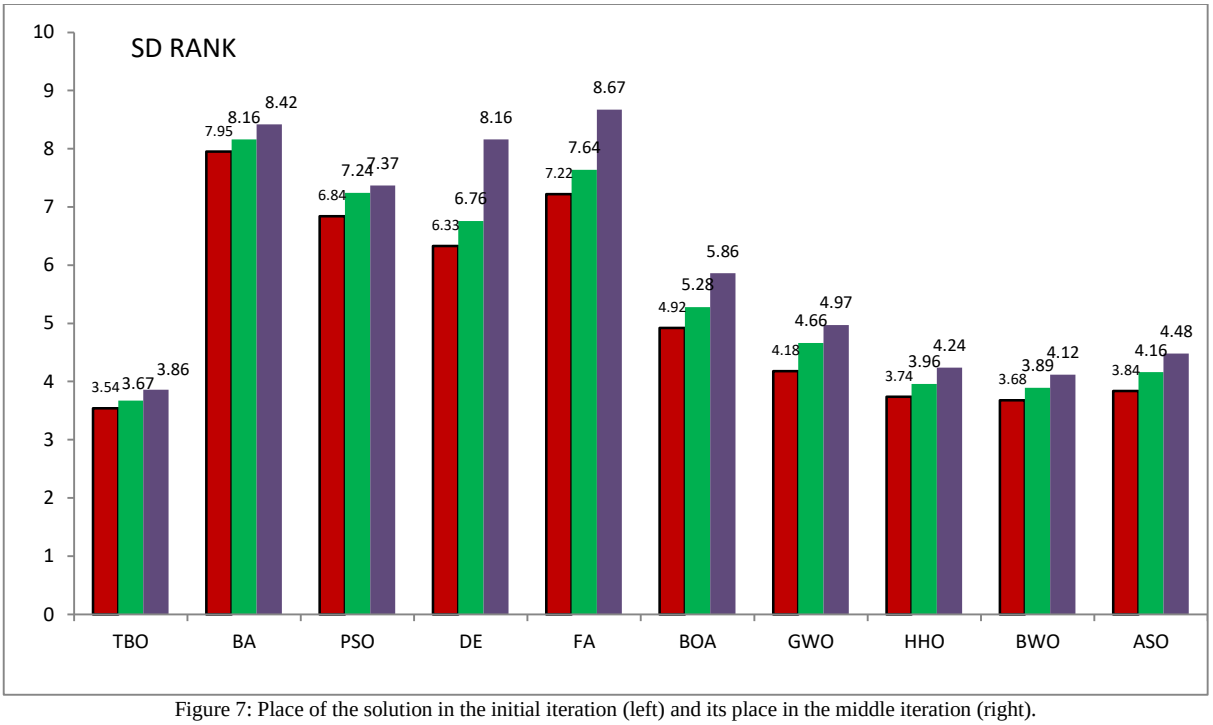

Figure 7: Place of the solution in the initial iteration (left) and its place in the middle iteration (right).

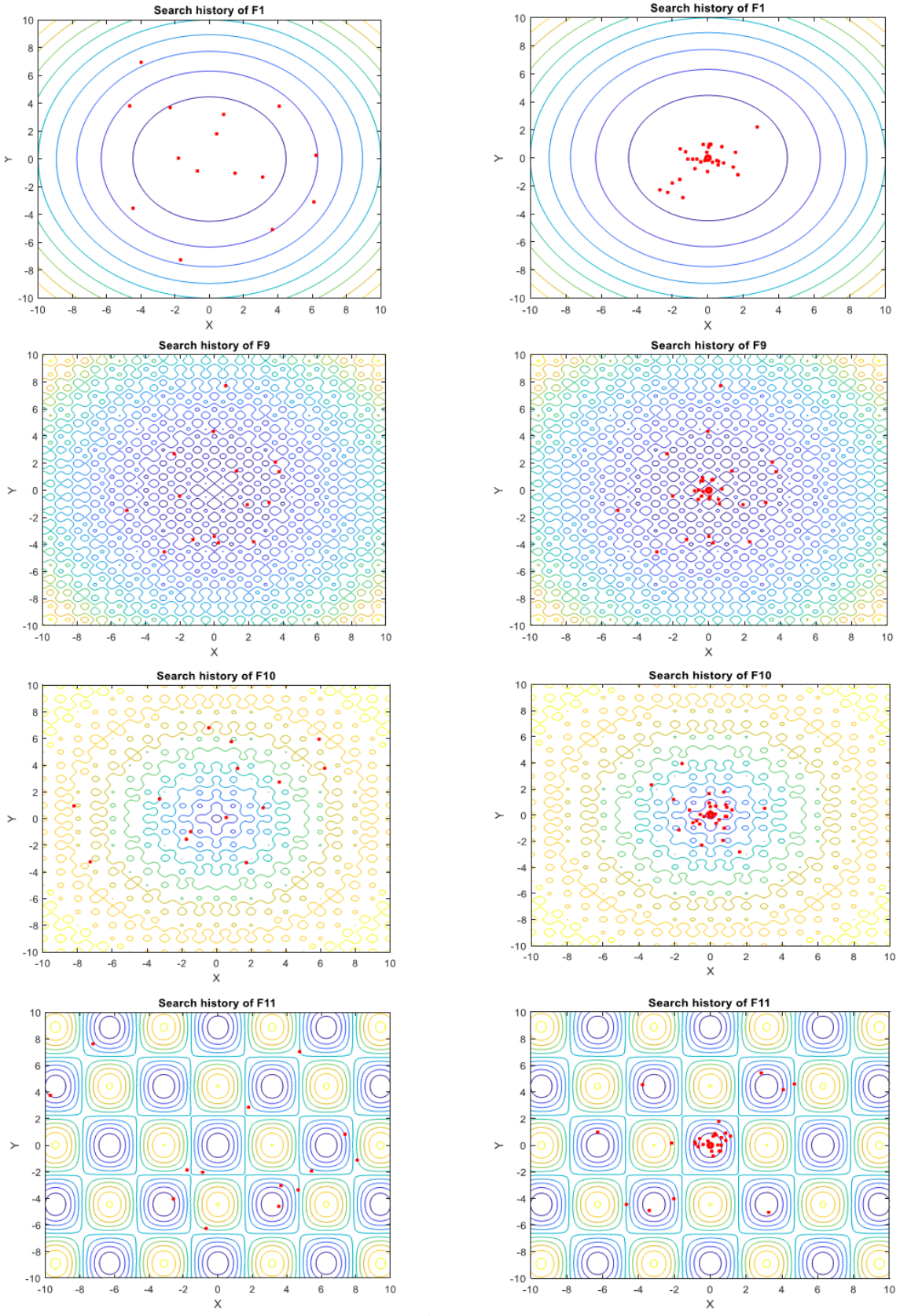

Figure 8: Initial iteration solution place (left) and place in succeeding middle iterations (right).

4.5 Analysing the worst solution

The iteration-based trajectory acts as a worthwhile index for assessing metaheuristic algorithms. Figure 9 shows the trajectory index [37] across iteration processes for evaluation functions F1, F9, F10, and F11. Examining the trajectory graphs demonstrates an advancement in the rate of the worst members and solutions within the Tiger Beetle algorithm. The space between the worst solution and the actual optimum decreases as the number of iterations increases. The convergence of the trajectory graph in a linear manner with zero ramps implies that the members of the worst population in the tiger beetle algorithm are strategically decreasing their distance from the optimal solution.

4.6 Analysis of fitness of the whole population

Figure 9 depicts the average fitness of the population associates across iterations in the TBO for evaluation functions F1, F9, F10, and F11. The investigation of middle fitness diagrams for all populations in the algorithm indicates a uniform decrease in movement, revealing a straight decrease in the average fitness. Since the objective function is of the minimization kind, the critical movement in this diagram illustrates progress in the algorithm's fitness.

Reducing the average error of the population members within the TBO recommends that the population effectively guide its members toward the global optimum, slowly reducing their distance from it.

5. REAL-WORL APPLICATION PROBLEMS

5.1 Cybersecurity and Data Protection

Among the sophisticated forms of malware are polymorphic viruses, which are adept at changing their behaviour while maintaining the same essential functions. We propose implementing a machine learning-based malware analysis system to counteract these evolving threats. This system comprises three integral modules: data processing, feature extraction, and malware detection. By leveraging machine learning capabilities, this approach aims to enhance the efficiency and accuracy of malware identification and mitigation strategies.

In contrast to the machine learning approach based on MPSO utilized in [49], our study opted for the proposed TBO to assess its performance in terms of accuracy, detection rate, and response time. We employed both honeypots deployed in real-world scenarios and publicly accessible benchmarked datasets to validate the proposed model utilized in [49].

In this study, we utilized TBO's fitness function to select the optimal number of features from each group. The main goal was to reduce errors in classification by calculating the balanced accuracy. With this in mind, the algorithms at the core of our study were utilized to perform the statistical analyses. Our primary focus was to evaluate the efficacy of the proposed TBO model compared to that of other modern models through experimentation. A specialized detection algorithm was designed to identify malware and was tested against both current and improved models. In Table 9, the performance metrics of our approach are compared to those of existing methods.

Table 9: PERFORMANCE METRICS OF THE PROPOSED METHOD COMPARED TO THOSE OF EXISTING METHODS

Detection Method	Detection Rate	Task Completion time	Accuracy
ML-MPSO [49]	84.3	98	89
ACO-LSTM[50]	89.1	101	87.9
RBBO[54]	82.7	98	90.7
Deep Learning [52]	88.1	106	92.5
ANN [53]	87.4	103	91.2
ML-TBO (Proposed)	**91.8**	**87**	**93.8**

According to Table 9, the ML-TBO method outperforms other detection methods in accurately identifying malware. With an impressive detection rate of 91.8%, it successfully targets a considerable number of malware instances, exhibiting its exceptional efficacy. This surpasses the performance of widely used methods such as ML-MPSO, ACO-LSTM, RBBO, deep learning, and ANN.

This approach is quite efficient since this task is processed in a very short duration of 87 seconds, hence signalling rapid and efficient processing. ACO-LSTM had shorter task completion times than did RBBO, deep learning, and ANN. Furthermore, ML-TBO achieves a very high accuracy of 93.8%, indicating its ability to classify malware samples with a very high proportion of precision. The proposed method outperforms the ML-MPSO, ACO-LSTM, RBBO, deep learning, and ANN methods in terms of accuracy.

5.2 Medical Image Segmentation

A quintessential application within the domain of medical image segmentation is the use of metaheuristic algorithms. Clustering methods have been reported that use these algorithms to interpret diseases such as brain tumors and breast cancer. Fuzzy clustering (FC) is used for segmentation. Its accuracy is appropriate for the selection of the number of cluster centers or the shape of clusters. However, there still has to be definite processes proposed for this selection. Clustering methods, including fuzzy clustering, surround direct clustering and insensitivity to image noise, have significant benefits. Unlike neural networks, these methods do not have a training phase.

A critical challenge in the FC algorithm for medical image segmentation of fibs is the optimal selection of cluster depths. The accuracy of medical image segmentation employing fuzzy clustering can be significantly improved by optimally selecting cluster centers. The nature of this optimization lies in the optimal selection of the fuzzy algorithm. Preparing medical image segmentation with fuzzy clustering as an optimization problem involves specifying the objective function as the error rate in image segmentation.

In this study, the TBO algorithm is employed to determine the optimal membership matrices to identify cluster centers in the fuzzy clustering algorithm for chest image segmentation. In this proposed method, specific membership matrices are ministered as solutions and considered members within the population of the tiger beetle algorithm. The algorithm endeavors to optimize the arrays within the membership matrices, employing the tiger beetle optimization algorithm to select more effective cluster centers. The proposed segmentation method is anticipated to exhibit increased accuracy by optimizing cluster center selection, enabling the identification of unhealthy tissue edges with reduced errors. In this method, each membership matrix is treated as a tiger beetle, as specified in Equation (13):

$$TB_i = [U_{n.c}]^i \tag{13}$$

The membership matrix consists of n rows illustrating pixels, while c represents the number of columns or clusters. In fuzzy clustering, the membership matrix is associated with TB_i and dined as a member within the tiger beetle population. Equation (14) describes the membership matrix as a population member in TBO:

$$TB_i = \begin{bmatrix} u_{1,1}^i & \dots & u_{1,c}^i \\ u_{2,1}^i & \dots & u_{2,c}^i \\ \dots & \dots & \dots \\ u_{n,1}^i & \dots & u_{n,c}^i \end{bmatrix}_{n \times c} \tag{14}$$

Each set of values in this matrix represents a fuzzy number, distinguishing the extent of a pixel's dependence on a cluster centre in similarity terms. The TBO algorithm and its interrelated equations focus on updating the membership matrices during each iteration. Utilizing a membership matrix or TB allows for the computation of interrelated cluster centers, as depicted in Equation (15).

$$c_i = \frac{\sum_{k=1}^{n} \mu_{ik}^m x_k}{\sum_{k=1}^{n} \mu_{ik}^m} \tag{15}$$

Every membership matrix or TB is assessed based on the existing condition of cluster centers utilizing the objective function of FC, as demonstrated in Equation (16) [25],[26]:

$$f(TB_i) = \sum_{k=1}^{n} \sum_{i=1}^{c} (u_{ik})^m \, d(x_k, c_i) \tag{16}$$

In this equation, $d(x_k, c_i)$ denotes the similarity of a pixel x_k to a cluster centre c_i. The similarity increases as the contrast in their light intensities decreases. Table 9 provides a visual representation of the flowchart for the proposed modified FC algorithm.

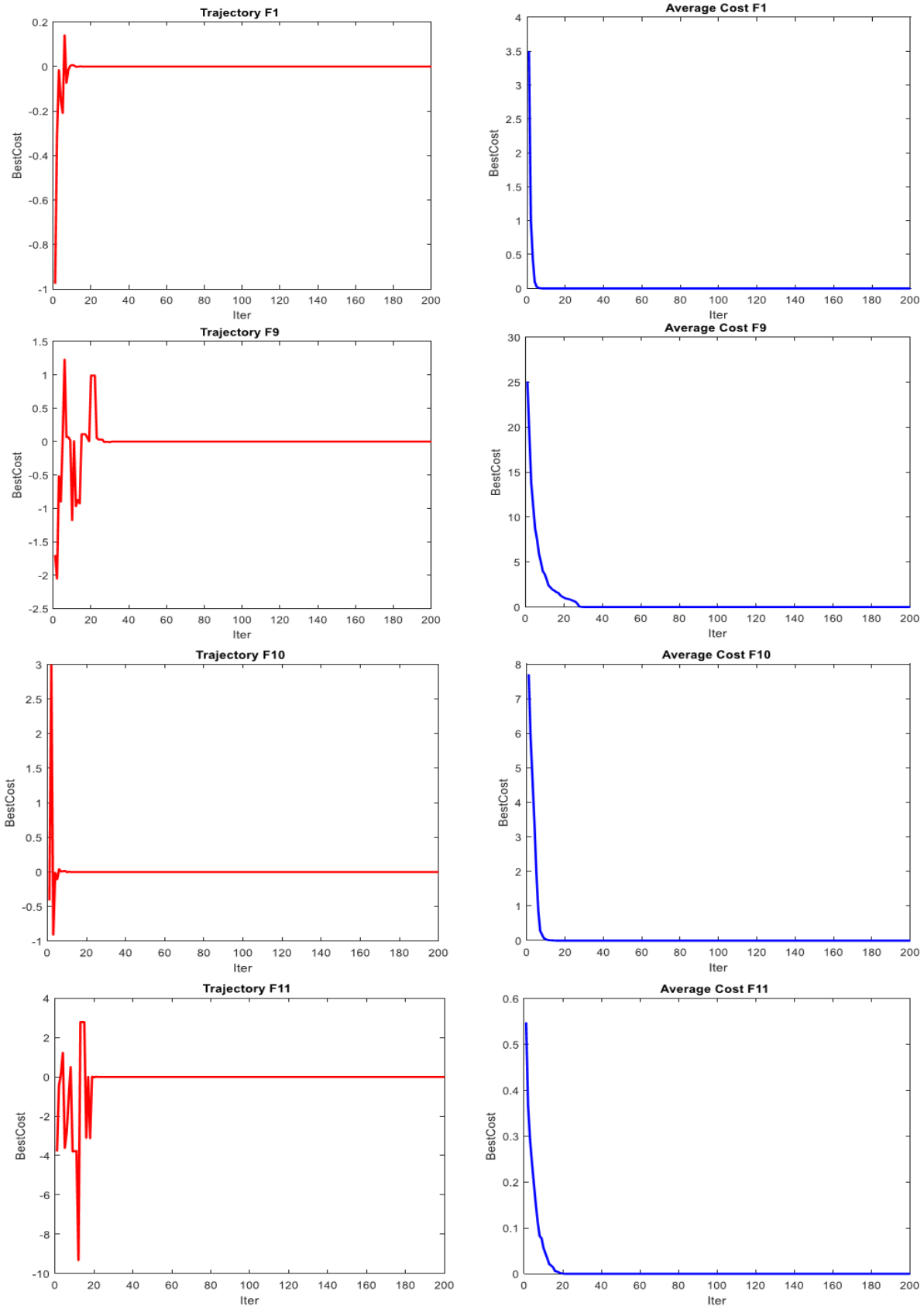

Figure 9: Solution position in the first iteration (left) and position in the middle iterations (right).

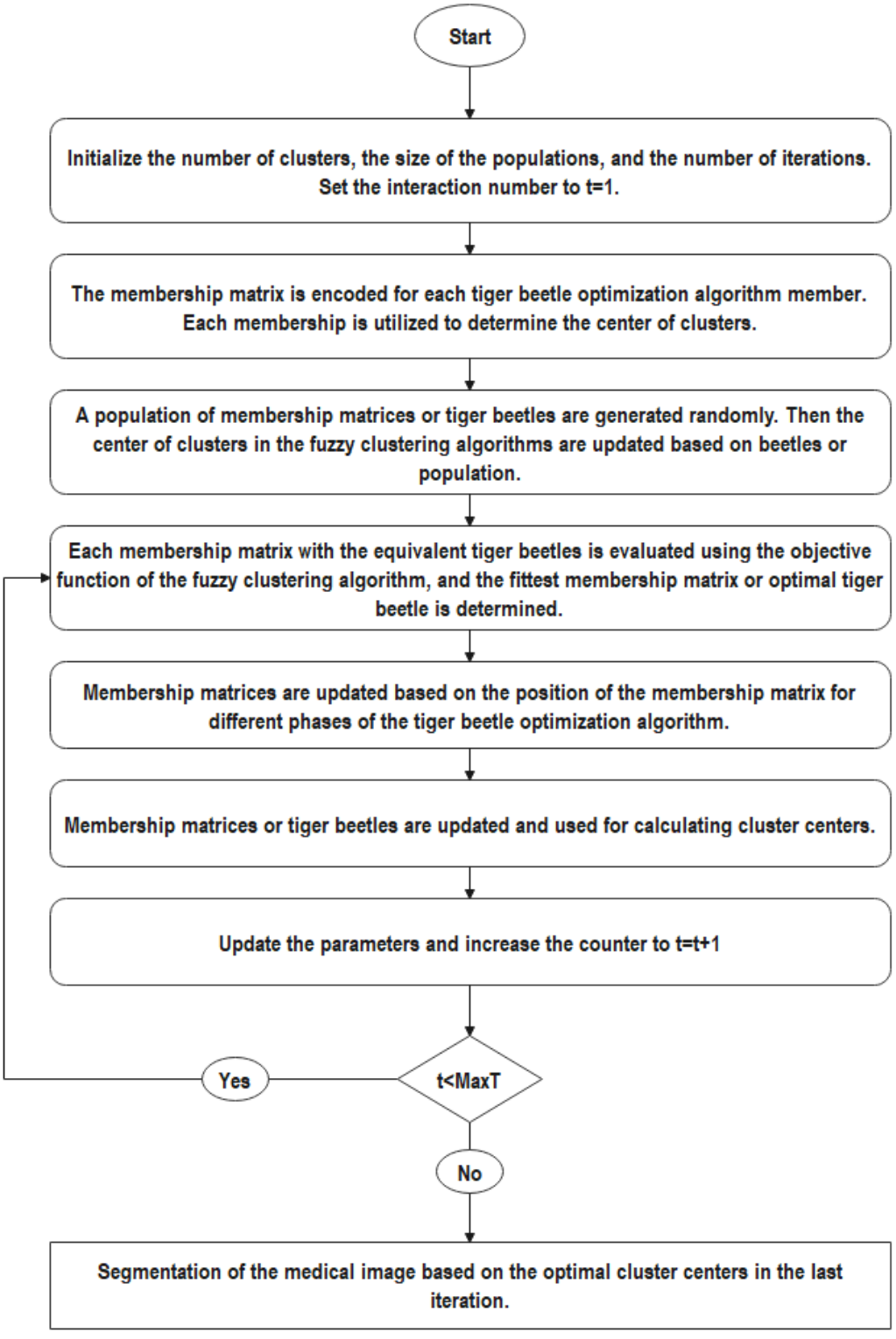

Figure 10: Suggested segmentation flowchart for image segmentation.

In the second stage of the proposed algorithm, the TBO algorithm was combined with fuzzy clustering for medical image segmentation. The segmentation algorithm can extract injured areas in medical imaging tissues, and preprocessing the image is needed before implementing the algorithm. After preprocessing, the segmentation algorithm can be used. In this context, two medical images, one showing brain tumors and the other showing lung CT scans, were considered.

1) Samples of medical images

This study employs two sets of images of brain tumors and breast cancer to implement the proposed image segmentation method. Figure *11* shows two samples of the brain tumor's medical images, and Figure *12* also displays a sample of breast cancer images. The images are gray, their light intensity channel is set between 0-255, and cancerous areas are also prominent.

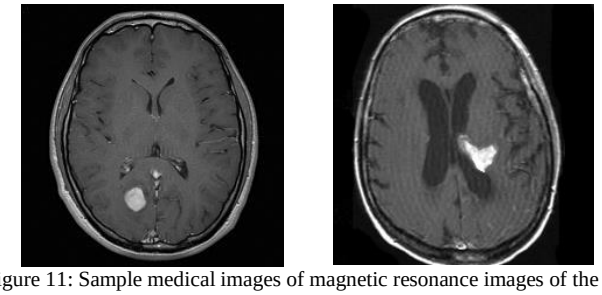

Figure 11: Sample medical images of magnetic resonance images of the brain.

2) Preprocessing medical example images

Brain magnetic resonance images are sensitive to noise, which can adversely influence image quality [38],[39]. To improve the accuracy of the segmentation method, it is crucial to first decrease noise before devoting the segmentation procedure. Noise reduction is vital because it maintains the precision of image segmentation, confirming that the tumor area is extracted with minimal error.

To reduce noise, a hybrid of two filters, namely, average and median filters, can be used, as delineated in Equations (17) and (18), respectively [28]:

$$f = \frac{1}{K}\sum_{i=1}^{k} MatrixImg_i(i,j) \tag{17}$$

$$f = \underset{(i,j) \in K}{median} \{MatrixImg(i,j)\} \tag{18}$$

Figure 12: Sample medical images of magnetic resonance images of the lung.

Here, k denotes the number of pixels neighboring the central pixel, MatrixImg indicates the medical image with noise, and f represents the coordinates of the image after the noise reduction procedure. The experimental results suggest that the median filter achieves better noise elimination within this algorithm.

3) Diagnosis using the tiger beetle algorithm

In Figure *11*and Figure *12*, the presented algorithm is shown via two magnetic resonance imaging samples for brain tumor diagnosis and two images for identifying damaged regions in lung tissue. The consequent output is a binary image, with the white areas displaying the damaged tissues.

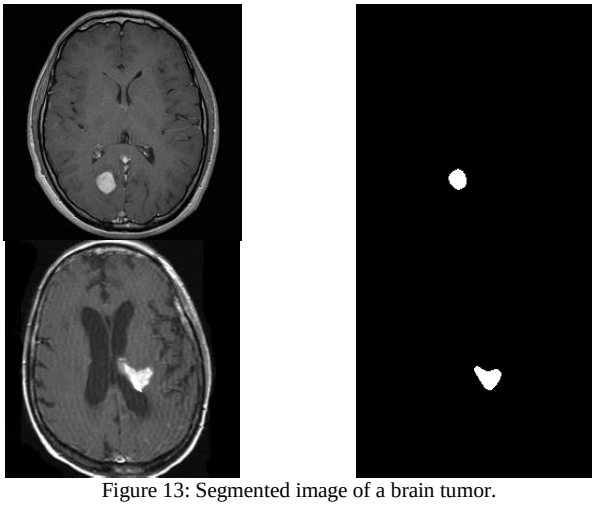

Figure 13: Segmented image of a brain tumor.

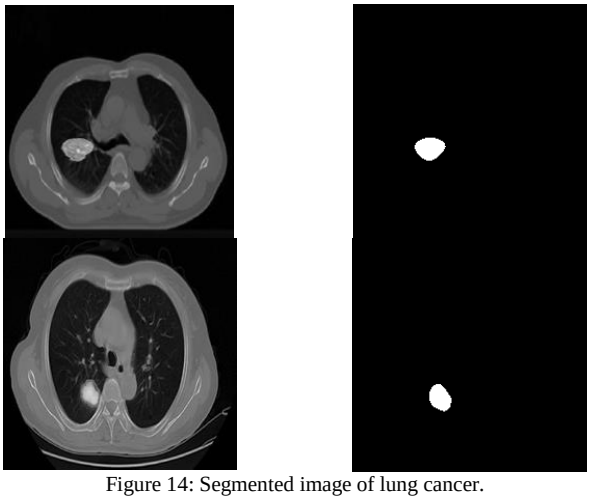

Figure 14: Segmented image of lung cancer.

Therefore, the proposed algorithm successfully pulls brain tumours or harmed lung tissue and accurately determines them in concurrence with the fuzzy clustering algorithm. The dataset from [30] can be employed to evaluate the efficacy of the proposed method in diagnosing lung cancer. The results of the proposed method for diagnosing lung cancer and the techniques used in this article are compared according to Table *10* and Figure *15*.

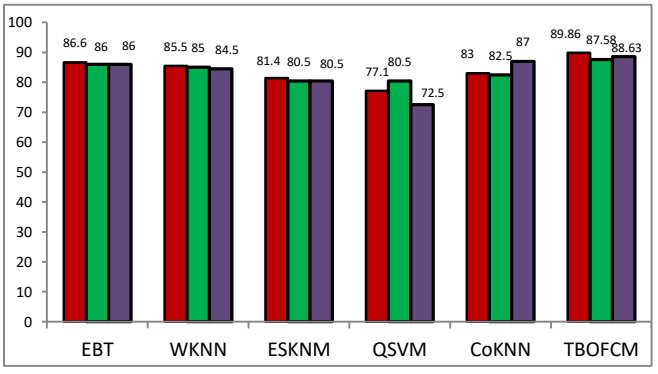

Figure 15: Evaluation of the proposed algorithm compared to existing lung cancer diagnosis methods.

Table 10: COMPARISON OF THE PROPOSED ALGORITHM WITH LUNG CANCER DIAGNOSIS METHODS

Algorithm	Sensitivity	Accuracy	Precision
EBT	86	86	86.6
WKNN	84.5	85	85.5
ESKNM	80.5	80.5	81.4
QSVM	72.5	80.5	77.1
CoKNN	87.0	82.5	83
TBOFCM	88.63	87.58	89.86

The proposed method is compared in terms of accuracy, sensitivity, and accuracy with lung cancer diagnostic methods. The sensitivity, accuracy, and precision of the proposed method are 88.63%, 87.58%, and 89.86%, respectively, and the proposed method performs better in diagnosing lung cancer than do the other methods.

5.3 Reservoir well placement

Well placement optimization is the process of determining the optimal locations for new wells in a reservoir. The goal of well placement optimization is to maximize the economic value of the reservoir by increasing oil and gas production. The PUNQ-S3 case, a synthetic reservoir model based on a real field, was selected to perform the required reservoir simulations for well placement optimization [46].

The net present value (NPV) is an objective function used to optimize well placement. The NPV can be calculated using the following equation (Equation (19)):

$$NPQ = \sum_{i=0}^{T} \frac{Q_o P_o - Q_w C_w - OPEX}{(1+D)^i} - CAPEX \quad (19)$$

The NPV of a well placement project is calculated by discounting the future cash flows from the project to the present day using a discount rate that reflects the time value of money and the risk of the project. The future cash flows are calculated based on the cumulative oil production (Q_o), cumulative water production Q_w), oil price (P_o), cost per unit volume of produced water C_w), operational expenditure (OPEX), and capital expenditure (CAPEX). The NPV is then used to optimize the well placement by finding the placement that maximizes the NPV.

The TBO consists of the following steps in terms of well placement optimization:

- Initialize the population as tiger beetles.
- Encode the holes dug in Equation (4) as wells.
- The hole positioning is performed through Equation (5).
- Update the hole's positions and kill the nonqualified tiger beetles via Equations (7) and (8).
- Iterate the algorithm optimum solution reached or maximum iteration number.
- Select the optimum solutions.

Optimization algorithms were used to maximize the NPV of the optimal placement of production wells. For all population-based algorithms, the initial population size was fixed at 25 tiger beetles and individuals. The parameter settings are adopted from [46]. Metaheuristic algorithms are powerful tools for solving complex optimization problems, but they are not guaranteed to converge to the global optimum. In most cases, metaheuristic algorithms can only find local optima.

However, if the algorithm is well designed and implemented, it is likely to find a good solution to the problem, even if it is not the global optimum. We compared the developed TBO algorithm with a recently proposed metaheuristic algorithm called DBO. Additionally, we compared TBO with PSO regarding the reservoir well placement problem. Figure *16* shows the general improvement in the NPV over successive iterations using TBO, PSO, and DBO. In all the algorithms, each iteration consists of 25 reservoir simulations. TBO is able to reach the optimal solution.

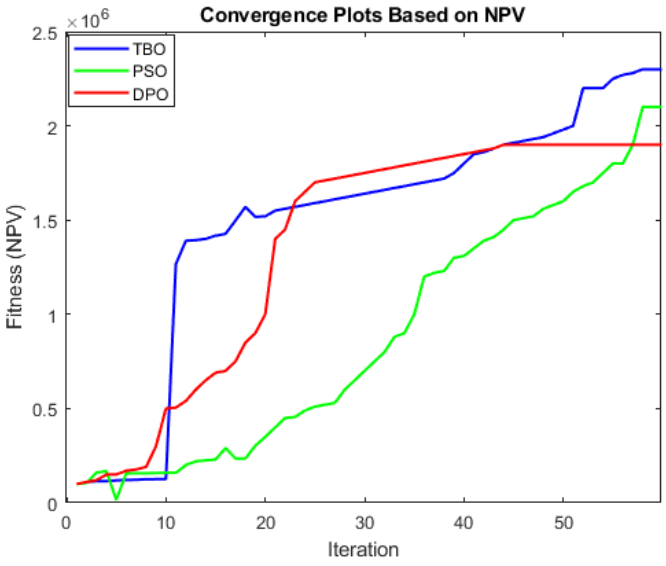

Figure 16: Convergence plots for TBO, DBO and PSO

6. CONCLUSIONS

In computer science, metaheuristic methods are utilized to solve problems. The behavior of living creatures and natural phenomena often inspires these algorithms. Tiger beetles, for instance, exhibit an intelligent hunting mechanism, making them an interesting subject of study. This research aimed to generate a metaheuristic method by emulating the behavior of tiger beetles. The result is the tiger beetle optimization algorithm, a population-based algorithm that mimics the search for prey or the optimal position of tiger beetles.

Through investigations with evaluation functions, it has been determined that the proposed algorithm excels in accurately specifying the global optimum approximated by various similar algorithms, including differential evolution, moth-flame, particle swarm, firefly, bat algorithm, gray wolf, black widow, Harris hawk, atom search algorithms, and the recently introduced DBO. Among its counterparts, the TBO algorithm always ranks first in most evaluation functions for finding optimal solutions. The advanced intelligence embedded in the tiger beetle algorithm ensures a reduced error rate and lower standard deviation (SD) when determined with alternative methods. Our suggested algorithm considerably benefits in maintaining robustness and precision, even when confronted with more challenging problems.

This study evaluated the error and convergence of recently presented metaheuristic algorithms for decoding a complex optimization problem. We found that the TBO algorithm had the best overall performance, with a lower average error and the fastest convergence rate. The DBO algorithm showed good performance but needed to be more accurate and efficient than the TBO algorithm. The results of the tests showed that the best overall time complexity was achieved by the TBO algorithm, followed by the PSO and DBO algorithms.

All the times in the presented table depend on the problem's size, and the TBO algorithm was the most scalable among the three. From our determinations, it is evident that the TBO algorithm is a very good solution for complex optimization problems, particularly with regard to accuracy and efficiency. However, there is still more work to be done to understand how TBO algorithms work from a wider view of issues and to develop new improved TBO algorithms.

From the statistical analysis of the results, it is concluded that the TBO algorithm is the most productive metaheuristic algorithm in comparison to PSO and DBO for solving the problem at hand. The reason for this is the search space and local optima conditions from which the TBO algorithm can properly escape.

Its good substantial devotion will encourage machine learning in malware detection with ML-Tjsonbo. The ML-Tjsonbo model yielded good performance metrics in terms of detection rate, completion of studies efficiently, and accuracy. ML-TBO can adapt to changes in malware, especially against conditions such as polymorphic viruses, making ML-TBO a strong solution in cybersecurity. Therefore, the results clearly indicate that ML-TBO has potential for improving the accuracy and efficiency of malware detection systems. This, in turn, leads to very valuable assets in the fight against

unwinding cyber threats. Additional research and implementation of ML-TBO are affirmed to capitalize on its stability and contribute to the ongoing improvement of cybersecurity measures.

We also assessed the performance of the TBO algorithm for decoding two real-world applications: image segmentation and reservoir well placements. We found that TBO outperformed the other algorithms in terms of accuracy and efficiency. As a practical application of the proposed method, the tiger beetle algorithm is used to optimize the fuzzy clustering algorithm in medical image segmentation. Furthermore, experiments have shown that the use of the tiger beetle algorithm combined with the fuzzy clustering algorithm results in highly accurate diagnoses of brain tumours and lung cancer. Our proposed method's sensitivity, accuracy, and precision in diagnosing lung cancer are 88.63%, 87.58%, and 89.86%, respectively, which surpasses the accuracy of lung cancer detection algorithms such as EBT, WKNN, ESKNM, QSVM, and CoKNN. In the future, we plan to use deep learning methods and tiger beetle optimization algorithms to diagnose lung cancer and achieve even greater accuracy. For reservoir well placements, TBO achieved a mean NPV of 2.6 million, while the other algorithms achieved NPVs of 2.4-2.5 million.

Our findings suggest that TBO is a promising algorithm for solving real-world problems, especially those where accuracy and efficiency are important. Some future research directions are as follows:

- The hybridization of TBO with other optimization algorithms is examined to improve its exploitation and exploration abilities.
- The application of TBO to other real-world problems, such as logistics and reserve chain optimization, renewable energy approach design, and cutting-edge material engineering, should be improved. Examining its performance in these disciplines can provide a more profound understanding of its applicable utility.
- Generate adaptive mechanisms for TBO parameters to enhance its robustness across various problem landscapes. Future research could concentrate on self-adaptive processes that dynamically change parameters in reaction to the optimization method, decreasing the demand for manual tuning.

On the high-dimensional multimodal functions F8 and F9-F13, the TBO algorithm outperforms the bat algorithm (BAT), PSO, WOA, BOA, and DBO algorithms, exhibiting better search ability. In fact, the TBO algorithm finds the global optimum on these functions. For function F8, the TBO algorithm ranks fourth after the WOA, DBO and HHO algorithms but still achieves competitive results compared to the BAT, PSO, WOA, BOA, and DBO algorithms. On function F13, the HHO algorithm outperforms the TBO algorithm, but the TBO algorithm remains competitive compared to other SI-based approaches.
Comparisons of results for low-dimensional functions are a valuable benchmark for testing new optimization algorithms and identifying promising candidates. Table \ref{table 6} shows the results of the TBO algorithm on low-dimensional functions, and it is notable that TBO performs well on F14-F18 and F20-23.

Funding

The authors had no institutional or sponsor backing.

Conflicts Of Interest

The author's disclosure statement confirms the absence of any conflicts of interest.

Acknowledgment

The authors extend appreciation to the institution for their unwavering support and encouragement during the course of this research.

References

[1] Rezaei, S. M., Pishvaee, H. A., Khosravi, A. R., & Jafari, M. A. (2023). A survey on metaheuristic optimization algorithms for resource allocation in fog computing environments. IEEE Access, 11, 3864622. doi:10.1109/ACCESS.2023.3864622.
[2] S. Mirjalili, J. S. Dong, A. S. Sadiq, and H. Faris, "Genetic algorithm: Theory, literature review, and application in image reconstruction", Spring Verlag. pp. 69-85, 2020.
[3] F. Padillo, J. M. Luna, and S. Ventura, "A grammar-guided genetic programming algorithm for associative classification in Big Data". Cognitive Computation, vol. 11, no.3, pp. 331-346, 2019.

[4] A. R. Verma, and B. Gupta, "A novel approach adaptive filtering method for electromyogram signal using Gray Wolf optimization algorithm", SN Applied Sciences, vol. 2, no.1, 16, 2020.
[5] A. Got, A. Moussaoui, and D. Zouache, "A guided population archive whale optimization algorithm for solving multiobjective optimization problems", Expert Systems with Applications, vol. 141,112972, 2020.
[6] V. Kumar, K. K. Kaleka, and Kaur, A, "Spiral-Inspired Spotted Hyena Optimizer and Its Application to Constraint Engineering Problems",Wireless Personal Communications, vol. 116, pp. 865-881, 2021.
[7] R. Sihwail, K. Omar, K. A. Z. Ariffin, and M. Tubishat, "Improved harris hawks optimization using elite opposition-based learning and novel search mechanism for feature selection", IEEE Access, vol. 8, pp. 121127-121145, 2020.
[8] V. Hayyolalam, and A. A. P. Kazem, "Black widow optimization algorithm: A novel meta-heuristic approach for solving engineering optimization problems", Engineering Applications of Artificial Intelligence, vol. 87, 103249, 2020.
[9] A. Kaveh, and A. D. Eslamlou, "Water strider algorithm: A new metaheuristic and applications", Structures, vol. 25, pp. 520-541, 2020.
[10] W. Al-Sorori, and A. M. Mohsen, "New Caledonian crow learning algorithm: A new metaheuristic algorithm for solving continuous optimization problems", Applied Soft Computing, vol 92, 106325, 2020.
[11] S. Ripon, M. G. Sarowar, F. Qasim, and S. T. Cynthia, "An Efficient Classification of Tuberous Sclerosis Disease Using Nature Inspired PSO and ACO Based Optimized Neural Network", In Nature Inspired Computing for Data Science, vol. 871, pp. 1-28, 2020.
[12] S. Ebadinezhad, "DEACO: Adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem", Engineering Applications of Artificial Intelligence, vol. 92, 103649, 2020.
[13] W. Zhao, L. Wang, and Z. Zhang, "Atom search optimization and its application to solve a hydrogeologic parameter estimation problem", Knowledge-Based Systems, vol. 163, pp. 283-304, 2019.
[14] A. Shabani, B. Asgarian, M. Salido, and S. A. Gharebaghi, "Search and rescue optimization algorithm: A new optimization method for solving constrained engineering optimization problems", Expert Systems with Applications, vol. 161, 113698, 2020.
[15] J. Ye, Y. Yu, Y. Zhang, Y. Todo, and S. Gao, "Improved Teaching-Learning-based Optimization Algorithm with Advanced Learning Strategy",Presented at 12th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2020
[16] S. H. S. Moosavi, and V. K. Bardsiri, "Poor and rich optimization algorithm: A new human-based and multi populations algorithm", Engineering Applications of Artificial Intelligence, vol. 86, pp. 165-181, 2019.
[17] E. Fadakar, and M. Ebrahimi, "A new metaheuristic football game inspired algorithm", Presented at 1st Conference on Swarm Intelligence and Evolutionary Computation (CSIEC), 2016.
[18] Ebadinezhad S (2020) DEACO: adopting dynamic evaporation strategy to enhance ACO algorithm for the traveling salesman problem. Eng Appl Artif Intel 92:103649
[19] Yang K, You X, Liu S, Pan H (2020) A novel ant colony optimization based on game for traveling salesman problem. Appl Intell 50(12):4529–4542
[20] Boussetta, M. S., Mekki, S., El-Sharkawi, M., & Younis, M. (2023). A hybrid metaheuristic algorithm for resource allocation in cloud computing environments. IEEE Access, 11, 10087-10099. doi:10.1109/ACCESS.2023.3802628.
[21] Z. M. Elgamal, N. B. M. Yasin, M. Tubishat, M. Alswaitti, and S. Mirjalili, "An Improved Harris Hawks Optimization Algorithm With Simulated Annealing for Feature Selection in the Medical Field", IEEE Access, vol. 8, pp. 186638-186652, 2020.
[22] ashaei, E., Pashaei, E. A fusion approach based on black hole algorithm and particle swarm optimization for image enhancement. Multimed Tools Appl 82, 297–325 (2023). https://doi.org/10.1007/s11042-022-13275-3
[23] Saihood, A., Karshenas, H., & Nilchi, A. R. N. (2022). Deep fusion of gray level co-occurrence matrices for lung nodule classification. PLOSONE, 17(9), e0274516. doi:10.1371/journal.pone.0274516
[24] X. Zhang, Y. Ding and Y. Liu, "A hybrid artificial bee colony algorithm for dynamic vehicle routing problem with time windows," in Journal of Intelligent Manufacturing, vol. 1, no. 1, pp. 273-287, 2022.
[25] S. Mirjalili, "SCA: A sine cosine algorithm for solving optimization problems", Knowledge-Based Systems, vol. 96, pp. 120-133, 2016.
[26] P. Mohindru, J. Kaur, and V. Akre, "Image Compression Using Hybrid Technique Combining VQ and Bacteria Foraging Optimization", International Journal of Advanced Research in Electrical Electronics and Instrumentation Engineering, vol. 9, no.1, pp. 2774-2780, 2020.
[27] S. K. Baliarsingh, S. Vipsita, and B. Dash (2020). "A new optimal gene selection approach for cancer classification using enhanced Jaya-based forest optimization algorithm", Neural Computing and Applications, vol. 32, no. 12, pp. 8599-8616, 2020.

[28] Z. Zhao, J. Zhao, K. Song, A. Hussain, Q. Du, Y. Dong, and X. Yang, X, "Joint DBN and Fuzzy C-Means unsupervised deep clustering for lung cancer patient stratification", Engineering Applications of Artificial Intelligence, vol. 91, 103571, 2020.
[29] A. B. Mathews, and M. K. Jeyakumar, "Analysis of Lung Tumor Detection using Various Segmentation Techniques", Presented at International Conference on Inventive Computation Technologies (ICICT), pp. 454-458, 2020.
[30] W. Xie, C. Xing, J. Wang, S. Guo, M. W. Guo, and L. F. Zhu, "Hybrid Henry Gas Solubility Optimization Algorithm Based on the Harris Hawk Optimization", IEEE Access, vol. 8, pp. 144665-144692, 2020.
[31] A. Krishna, P. S. Rao, and C. Z. Basha, "Computerized Classification of CT Lung Images using CNN with Watershed Segmentation", Presented at Second International Conference on Inventive Research in Computing Applications (ICIRCA), pp. 18-21, 2020.
[32] X. Xu, C. Wang, J. Guo, Y. Gan, J. Wang, H. Bai, and Z. Yi, Z, "MSCS-DeepLN: Evaluating lung nodule malignancy using multi-scale costsensitive neural networks", Medical Image Analysis, vol. 65, 101772, 2020.
[33] M. A. Khan, S. Rubab, A. Kashif, M. I. Sharif, N. Muhammad, J. H. Shah, and S. C. Satapathy, "Lungs cancer classification from CT images: An integrated design of contrast based classical features fusion and selection", Pattern Recognition Letters, vol. 129, pp. 77-85, 2020.
[34] N. Dhanachandra, and Y. J. Chanu,"An image segmentation approach based on fuzzy c-means and dynamic particle swarm optimization algorithm", Multimedia Tools and Applications, vol. 79, pp.18839-18858, 2020.
[35] J. Arora, and M. Tushir, "Hybrid KFCM-PSO Clustering Technique for Image Segmentation", Presented at Proceedings of International Conference on Artificial Intelligence and Applications, pp. 443-451, 2020.
[36] X. Chu, F. Cai, D. Gao, L. Li, J. Cui, S. X. Xu, and Q. Qin, "An artificial bee colony algorithm with adaptive heterogeneous competition for global optimization problems", Applied Soft Computing, vol. 93, 106391, 2020.
[37] V. K. Kamboj, A. Nandi, A. Bhadoria, and S. Sehgal, "An intensify Harris Hawks optimizer for numerical and engineering optimization problems", Applied Soft Computing, vol. 89, 106018, 2020.
[38] J. Jiang, R. Jiang, X. Meng, K. Li, "SCGSA: A sine chaotic gravitational search algorithm for continuous optimization problems", Expert Systems with Applications, vol. 144, 113118, 2020.
[39] Z. K. Feng, W. J. Niu, and S. Liu, "Cooperation search algorithm: A novel metaheuristic evolutionary intelligence algorithm for numerical optimization and engineering optimization problems", Applied Soft Computing, vol. 98, 106734, 2020.
[40] R. Chai, A. Savvaris, A. Tsourdos, and S. Chai, "Overview of Trajectory Optimization Techniques. In Design of Trajectory Optimization Approach for Space Maneuver Vehicle Skip Entry Problems", Springer Singapore, pp. 7-25, 2020.
[41] N. Abiwinanda, M. Hanif, S. T. Hesaputra, A. Handayani, and T. R. Mengko, (2019). "Brain tumour classification using convolutional neural network", Presented at World Congress on Medical Physics and Biomedical Engineering, Springer Singapore, pp. 183-189, 2018.
[42] Li, Ping, S. Wang, T. Li, J. Lu, Y. HuangFu, and D. Wang. "A large-scale CT and PET/CT dataset for lung cancer diagnosis [dataset]." The cancer imaging archive (2020).
[43] Siemanowski J, Heydt C, Merkelbach-Bruse S. Predictive molecular pathology of lung cancer in Germany with focus on gene fusion testing: Methods and quality assurance. Cancer Cytopathol. 2020 Sep;128(9):611-621. doi: 10.1002/cncy.22293. PMID: 32885916.
[44] Xue J, Shen B. Dung beetle optimizer: a new meta-heuristic algorithm for global optimization [Internet]. Vol. 79, Journal of Supercomputing. Springer US; 2023. 7305–7336 p. Available from: https://doi.org/10.1007/s11227-022-04959-6
[45] Naderi M, Khamehchi E. Well placement optimization using metaheuristic bat algorithm. J Pet Sci Eng [Internet]. 2017;150:348–54. Available from: http://dx.doi.org/10.1016/j.petrol.2016.12.028
[46] Raji S, Dehnamaki A, Somee B, Mahdiani MR. A new approach in well placement optimization using metaheuristic algorithms. J Pet Sci Eng [Internet]. 2022;215:110640. Available from: https://www.sciencedirect.com/science/article/pii/S0920410522005113
[47] M. M. Abubakar, A. Z. Umar and M. Abubakar, "Personal Data and Privacy Protection Regulations: State of compliance with Nigeria Data Protection Regulations (NDPR) in Ministries, Departments, and Agencies (MDAs)," 2022 5th Information Technology for Education and Development (ITED), Abuja, Nigeria, 2022, pp. 1-6, doi: 10.1109/ITED56637.2022.10051182.
[48] Z. Mao, HuiLi, Z. Huang, Y. Tian, X. Zhao and H. Zhang, "Ghostwriting-Federal Learning Key Technology Research for Big Data Privacy Protection," 2022 4th International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), Shanghai, China, 2022, pp. 387-390, doi: 10.1109/MLBDBI58171.2022.00080. keywords: Differential privacy;Analytical models;Machine learning algorithms;Publishing;Data protection;Big Data;Data models;Federated learning algorithms;big data privacy protection;anomalous data;defense mechanisms,

[49] S. Sivakumar, S. Saminathan, R. Ranjana, M. Mohan and P. K. Pareek, "Malware Detection Using The Machine Learning Based Modified Partial Swarm Optimization Approach," 2023 International Conference on Applied Intelligence and Sustainable Computing (ICAISC), Dharwad, India, 2023, pp. 1-5, doi: 10.1109/ICAISC58445.2023.10199796. keywords: Machine learning algorithms;Machine learning;Benchmark testing;Feature extraction;Malware;Particle swarm optimization;Principal component analysis;Machine Learning;Malware Detection;Particle Swarm Optimization (PSO);Optimal Solutions,

[50] J. Lande, M. Maheswari, S. M. Kamali, R. Dineshkumar and P. M. S, "Hybrid Optimization Based Long Short-Term Memory for Android Malware Detection," 2023 International Conference on Evolutionary Algorithms and Soft Computing Techniques (EASCT), Bengaluru, India, 2023, pp. 1-5, doi: 10.1109/EASCT59475.2023.10393408. keywords: Measurement;Ant colony optimization;Feature extraction; Malware;Libraries;Cleaning;Whale optimization algorithms;Ant Lion Optimization;Long Short-Term Memory;Malware Detection;NumPy and Particle Swarm Optimization,

[51] A. Jain, K. Tripathi, A. Jatain and Manju, "Anomaly Detection in the Cloud Environment with Clustering Optimization Model for Attack Detection in IDs," 2023 International Conference on IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India, 2023, pp. 1-5, doi: 10.1109/ICICAT57735.2023.10263676. keywords: Cloud computing;Intrusion detection;Estimation;Real-time systems;Data models;Bayes methods;Task analysis;Intrusion Detection System (IDS);Bayesian Network;Deep Learning;Optimization;Malicious Activities,

[52] P. Kotian and R. Sonkusare, "Detection of Malware in Cloud Environment using Deep Neural Network," 2021 6th International Conference for Convergence in Technology (I2CT), Maharashtra, India, 2021, pp. 1-5, doi: 10.1109/I2CT51068.2021.9417901. keywords: Deep learning;Cloud computing;Portable computers;Standards organizations;Standardization;Filtering algorithms;Malware;Convolutional Neural Networks;Malware detection;Deep Learning;Security,

[53] L. P. Khan, "Obfuscated Malware Detection Using Artificial Neural Network (ANN)," 2023 Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT), Erode, India, 2023, pp. 1-5, doi: 10.1109/ICECCT56650.2023.10179639. keywords: Electric potential;Costs;Computer hacking;Computational modeling;Organizations;Artificial neural networks;Learning (artificial intelligence);Malware;ANN;accuracy;precision;obfuscated;machine learning,

[54] A. Jain, K. Tripathi, A. Jatain and Manju, "Anomaly Detection in the Cloud Environment with Clustering Optimization Model for Attack Detection in IDs," 2023 International Conference on IoT, Communication and Automation Technology (ICICAT), Gorakhpur, India, 2023, pp. 1-5, doi: 10.1109/ICICAT57735.2023.10263676. keywords: Cloud computing;Intrusion detection;Estimation;Real-time systems;Data models;Bayes methods;Task analysis;Intrusion Detection System (IDS);Bayesian Network;Deep Learning;Optimization;Malicious Activities.